

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 NoDriver.php	27
E OTS NotAContainer.php	28
E OTS NotLoaded.php	29
E OTS OTBMEError.php	30
E OTS OutOfBuffer.php	31
E OTS ReadOnly.php	32
IOTS DAO.php	33
IOTS DB.php	34
IOTS FileCache.php	35
IOTS GuildAction.php	36
IOTS ItemsCache.php	37
OTS.php	38
OTS Account.php	39
OTS Accounts List.php	40
OTS Base DAO.php	41
OTS Base List.php	42
OTS Container.php	43
OTS DB MySQL.php	44
OTS DB ODBC.php	45
OTS DB PostgreSQL.php	46
OTS DB SQLite.php	47
OTS FileLoader.php	48
OTS FileNode.php	49
OTS Group.php	50
OTS Groups List.php	51
OTS Guild.php	52
OTS GuildRank.php	53
OTS GuildRanks List.php	54

OTS Guilds List.php	55
OTS House.php	56
OTS HousesList.php	57
OTS InfoRespond.php	58
OTS Item.php	59
OTS ItemsList.php	60
OTS ItemType.php	61
OTS MapCoords.php	62
OTS Monster.php	63
OTS MonstersList.php	64
OTS OTBMFile.php	65
OTS Player.php	66
OTS Players List.php	67
OTS Spell.php	68
OTS SpellsList.php	69
OTS SQLField.php	70
OTS SQLFilter.php	71
OTS SQLite Results.php	72
OTS VocationsList.php	73
Package POT Classes	74
Class E OTS ErrorCode	74
Constructor construct	74
Class E OTS FileLoaderError	75
Class Constant ERROR_CAN_NOT_OPEN	75
Class Constant ERROR_EOF	75
Class Constant ERROR_INVALID_FILE_VERSION	76
Class Constant ERROR_INVALID_FORMAT	76
Class Constant ERROR_NOT_OPEN	76
Class Constant ERROR_SEEK_ERROR	77
Class Constant ERROR_TELL_ERROR	77
Class E OTS NoDriver	78
Class E OTS NotAContainer	78
Class E OTS NotLoaded	79
Class E OTS OTBMErrors	79
Class Constant LOADMAPERROR_OUTDATEDHEADER	79
Class Constant LOADMAPERROR_UNKNOWNNODETYPE	80
Class E OTS OutOfBuffer	80
Class E OTS ReadOnly	81
Class IOTS DAO	81
Class IOTS DB	81
Constructor construct	82
Method fieldName	82
Method lastInsertId	83
Method limit	83
Method SQLquery	84
Method SQLquote	84
Method tableName	85
Class IOTS FileCache	85
Method readCache	85

Method writeCache	86
Class IOTS_GuildAction	87
Constructor __construct	87
Method addRequest	88
Method deleteRequest	88
Method listRequests	88
Method submitRequest	89
Class IOTS_ItemsCache	89
Method readItems	90
Method writeItems	90
Class OTS_Account	91
Method ban	91
Method block	92
Method count	92
Method create	93
example: account.php	93
Method createEx	94
Method delete	95
Method find	95
Method getCustomField	95
Method getEmail	96
Method getGroup	97
Method getId	97
Method getIterator	97
Method getPACCDays	98
Method getPassword	98
Method getPlayers	99
Method getPlayersList	99
Method isBanned	100
Method isBlocked	100
Method isLoaded	100
Method load	101
Method save	101
Method setCustomField	102
Method setEmail	102
Method setGroup	103
Method setPACCDays	103
Method setPassword	104
Method unban	104
Method unblock	105
Method __get	105
Method __set	106
Class OTS_Accounts_List	106
Method deleteAccount	107
Method init	107
Class OTS_Base_DAO	108
Var \$db	108
Constructor __construct	108
Method __clone	109

Method set_state	109
Method sleep	110
Method wakeup	110
Class OTS_Base_List	110
Var \$class	111
Var \$table	111
Constructor construct	112
Method count	112
Method current	112
Method init	113
Method key	113
Method next	114
Method orderBy	114
Method resetOrder	114
Method rewind	115
Method setFilter	115
Method setLimit	115
Method setOffset	116
Method valid	116
Method set	117
Method set_state	117
Method sleep	118
Method wakeup	118
Class OTS_Container	119
Method addItem	119
Method count	119
Method current	120
Method getIterator	120
Method key	121
Method next	121
Method removeItem	121
Method rewind	122
Method valid	122
Class OTS_DB_MySQL	123
Constructor construct	123
Method fieldName	124
Method limit	124
Method SQLquery	125
Method SQLquote	125
Method tableName	126
Class OTS_DB_ODBC	126
Constructor construct	127
Method fieldName	127
Method limit	128
Method SQLquery	128
Method SQLquote	129
Method tableName	129
Class OTS_DB_PostgreSQL	130
Constructor construct	130

Method fieldName	131
Method limit	131
Method SQLquery	132
Method SQLquote	132
Method tableName	133
Class OTS_DB_SQLite	134
Constructor construct	134
Method fieldName	135
Method limit	135
Method SQLquery	135
Method SQLquote	136
Method tableName	137
Class OTS_FileLoader	137
Class Constant ESCAPE_CHAR	138
Class Constant NODE_END	138
Class Constant NODE_START	138
Var \$cache	139
Var \$root	139
Method loadFile	139
Method setCacheDriver	140
Method clone	140
Method set	141
Method set_state	141
Method sleep	142
Class OTS_FileNode	142
Method getBuffer	143
Method getChar	143
Method getChild	144
Method getLong	144
Method getNext	144
Method getShort	145
Method getString	145
Method getType	145
Method isValid	146
Method setBuffer	146
Method setChild	147
Method setNext	147
Method setType	147
Method skip	148
Method clone	148
Method get	149
Method set	149
Method set_state	150
Class OTS_Group	150
Method count	151
Method delete	152
Method getAccess	152
Method getCustomField	152
Method getFlags	153

Method getId	153
Method getIterator	154
Method getMaxDepotItems	154
Method getMaxVIPList	155
Method getName	155
Method getPlayers	155
Method getPlayersList	156
Method isLoaded	156
Method load	157
Method save	157
Method setAccess	157
Method setCustomField	158
Method setFlags	159
Method setMaxDepotItems	159
Method setMaxVIPList	160
Method setName	160
Method get	160
Method set	161
Class OTS_Groups_List	162
Method deleteGroup	162
Method init	163
Class OTS_Guild	163
Method acceptInvite	164
Method acceptRequest	164
Method count	165
Method delete	165
Method deleteInvite	166
Method deleteRequest	166
Method find	167
Method getCreationData	167
Method getCustomField	167
Method getGuildRanks	168
Method getGuildRanksList	169
Method getId	169
Method getIterator	169
Method getName	170
Method getOwner	170
Method invite	171
Method isLoaded	171
Method listInvites	171
Method listRequests	172
Method load	172
Method request	173
Method save	173
Method setCreationData	174
Method setCustomField	174
Method setInvitesDriver	175
Method setName	175
Method setOwner	176

Method setRequestsDriver	176
Method clone	177
Method get	177
Method set	177
Method sleep	178
Class OTS_GuildRank	179
Method count	179
Method delete	179
Method find	180
Method getCustomField	180
Method getGuild	181
Method getId	181
Method getIterator	182
Method getLevel	182
Method getName	183
Method getPlayers	183
Method getPlayersList	183
Method isLoaded	184
Method load	184
Method save	185
Method setCustomField	185
Method setGuild	186
Method setLevel	186
Method setName	187
Method get	187
Method set	188
Class OTS_GuildRanks_List	188
Method deleteGuildRank	189
Method init	189
Class OTS_Guilds_List	190
Method deleteGuild	190
Method init	191
Class OTS_House	191
Constructor construct	192
Method addTile	192
Method delete	193
Method getEntry	193
Method getId	193
Method getName	194
Method getOwner	194
Method getPaid	194
Method getRent	195
Method getSize	195
Method getTiles	195
Method getTownId	196
Method getTownName	196
Method getWarnings	196
Method save	196
Method setOwner	197

Method setPaid	197
Method setWarnings	198
Method get	198
Method set	199
Method sleep	199
Class OTS HousesList	200
Constructor construct	200
Method count	200
Method getHouse	201
Method getHouseId	201
Method getIterator	202
Method offsetExists	202
Method offsetGet	202
Method offsetSet	203
Method offsetUnset	203
Method set state	204
Class OTS InfoRespond	205
Method getClientVersion	206
Method getEmail	206
Method getIP	206
Method getLocation	207
Method getMapAuthor	207
Method getMapHeight	207
Method getMapName	208
Method getMapWidth	208
Method getMaxPlayers	208
Method getMonstersCount	209
Method getMOTD	209
Method getName	209
Method getOnlinePlayers	210
Method getOwner	210
Method getPlayersPeak	210
Method getPort	210
Method getServer	211
Method getServerVersion	211
Method getTSPQVersion	211
Method getUptime	212
Method getURL	212
Method get	212
Class OTS Item	213
Constructor construct	214
Method count	214
Method getAttributes	214
Method getCount	215
Method getId	215
Method getItemType	215
Method setAttributes	216
Method setCount	216
Method get	217

Class OTS ItemsList	217
Class Constant CLIENT_VERSION 750	218
Class Constant CLIENT_VERSION 755	218
Class Constant CLIENT_VERSION 760	219
Class Constant CLIENT_VERSION 770	219
Class Constant CLIENT_VERSION 780	219
Class Constant CLIENT_VERSION 790	220
Class Constant CLIENT_VERSION 792	220
Class Constant CLIENT_VERSION 800	220
Class Constant ITEM_ATTR_CLIENTID	221
Class Constant ITEM_ATTR_LIGHT2	221
Class Constant ITEM_ATTR_SERVERID	221
Class Constant ITEM_ATTR_SPEED	222
Class Constant ITEM_ATTR_TOPORDER	222
Class Constant ROOT_ATTR_VERSION	222
Method count	223
Method current	223
Method getBuildVersion	223
Method getClientVersion	224
Method getItemType	224
Method getItemTypeId	224
Method getItemTypesList	225
Method getIterator	225
Method getOTBVersion	226
Method key	226
Method loadItems	226
Method next	227
Method offsetExists	227
Method offsetGet	228
Method offsetSet	228
Method offsetUnset	229
Method rewind	229
Method valid	230
Method get	230
Method set_state	231
Method wakeup	231
Class OTS ItemType	232
Class Constant FLAG_ALLOWDISTREAD	233
Class Constant FLAG_ALWAYSONTOP	233
Class Constant FLAG_BLOCK_PATHFIND	233
Class Constant FLAG_BLOCK_PROJECTILE	234
Class Constant FLAG_BLOCK_SOLID	234
Class Constant FLAG_CANNOTDECAY	234
Class Constant FLAG_FLOORCHANGEDOWN	235
Class Constant FLAG_FLOORCHANGEEST	235
Class Constant FLAG_FLOORCHANGENORTH	235
Class Constant FLAG_FLOORCHANGESOUTH	236
Class Constant FLAG_FLOORCHANGEWEST	236
Class Constant FLAG_HANGABLE	236

Class Constant FLAG HAS HEIGHT	237
Class Constant FLAG HORIZONTAL	237
Class Constant FLAG MOVEABLE	237
Class Constant FLAG PICKUPABLE	238
Class Constant FLAG READABLE	238
Class Constant FLAG ROTABLE	238
Class Constant FLAG STACKABLE	239
Class Constant FLAG USEABLE	239
Class Constant FLAG VERTICAL	239
Class Constant ITEM GROUP AMMUNITION	240
Class Constant ITEM GROUP ARMOR	240
Class Constant ITEM GROUP CONTAINER	240
Class Constant ITEM GROUP DOOR	241
Class Constant ITEM GROUP FLUID	241
Class Constant ITEM GROUP GROUND	241
Class Constant ITEM GROUP KEY	242
Class Constant ITEM GROUP MAGICFIELD	242
Class Constant ITEM GROUP NONE	242
Class Constant ITEM GROUP RUNE	243
Class Constant ITEM GROUP SPLASH	243
Class Constant ITEM GROUP TELEPORT	244
Class Constant ITEM GROUP WEAPON	244
Class Constant ITEM GROUP WRITEABLE	244
Class Constant ITEM TYPE CONTAINER	245
Class Constant ITEM TYPE DEPOT	245
Class Constant ITEM TYPE DOOR	245
Class Constant ITEM TYPE MAGICFIELD	246
Class Constant ITEM TYPE MAILBOX	246
Class Constant ITEM TYPE NONE	246
Class Constant ITEM TYPE TELEPORT	247
Class Constant ITEM TYPE TRASHHOLDER	247
Constructor <code>__construct</code>	247
Method <code>createItem</code>	248
Method <code>getAttribute</code>	248
Method <code>getAttributesList</code>	249
Method <code>getClientId</code>	249
Method <code>getGroup</code>	249
Method <code>getId</code>	250
Method <code>getName</code>	250
Method <code>getType</code>	250
Method <code>hasHeight</code>	250
Method <code>isAlwaysOnTop</code>	251
Method <code>isBlocking</code>	251
Method <code>isHangable</code>	251
Method <code>isHorizontal</code>	252
Method <code>isMovable</code>	252
Method <code>isPickupable</code>	252
Method <code>isReadable</code>	253
Method <code>isRotable</code>	253

Method isStackable	253
Method isUsable	254
Method isVertical	254
Method setAttribute	254
Method setClientId	255
Method setFlags	255
Method setGroup	256
Method setName	256
Method setType	257
Method get	257
Method set	258
Method set state	258
Class OTS_MapCoords	259
Constructor construct	259
Method getX	260
Method getY	260
Method getZ	261
Method get	261
Method set state	261
Class OTS_Monster	262
Method getArmor	263
Method getAttacks	263
Method getDefense	264
Method getDefenses	264
Method getExperience	264
Method getFlag	265
Method getFlags	265
Method getHealth	265
Method getImmunities	266
Method getItems	266
Method getLoot	266
Method getManaCost	267
Method getName	267
Method getRace	267
Method getSpeed	268
Method getVoices	268
Method hasImmunity	268
Method get	269
Class OTS_MonstersList	270
Constructor construct	270
Method count	270
Method current	271
Method getMonster	271
Method key	271
Method next	272
Method offsetExists	272
Method offsetGet	272
Method offsetSet	273
Method offsetUnset	273

Method rewind	274
Method valid	274
Method set state	275
Class OTS OTBMFile	275
Class Constant OTBM ATTR ACTION ID	276
Class Constant OTBM ATTR DEPOT ID	276
Class Constant OTBM ATTR DESC	276
Class Constant OTBM ATTR DESCRIPTION	277
Class Constant OTBM ATTR EXT FILE	277
Class Constant OTBM ATTR EXT HOUSE FILE	277
Class Constant OTBM ATTR EXT SPAWN FILE	278
Class Constant OTBM ATTR HOUSEDORID	278
Class Constant OTBM ATTR ITEM	279
Class Constant OTBM ATTR RUNE CHARGES	279
Class Constant OTBM ATTR TELE DEST	279
Class Constant OTBM ATTR TEXT	280
Class Constant OTBM ATTR TILE FLAGS	280
Class Constant OTBM ATTR UNIQUE ID	280
Class Constant OTBM NODE HOUSETILE	281
Class Constant OTBM NODE ITEM	281
Class Constant OTBM NODE ITEM DEF	281
Class Constant OTBM NODE MAP DATA	282
Class Constant OTBM NODE MONSTER	282
Class Constant OTBM NODE ROOTV1	282
Class Constant OTBM NODE SPAWNS	283
Class Constant OTBM NODE SPAWN AREA	283
Class Constant OTBM NODE TILE	283
Class Constant OTBM NODE TILE AREA	284
Class Constant OTBM NODE TILE REF	284
Class Constant OTBM NODE TILE SQUARE	284
Class Constant OTBM NODE TOWN	285
Class Constant OTBM NODE TOWNS	285
Method count	285
Method current	286
Method getDescription	286
Method getHeight	286
Method getHousesList	287
Method getIterator	287
Method getTownID	287
Method getTownName	288
Method getTownsList	288
Method getTownTemple	289
Method getWidth	289
Method key	290
Method loadFile	290
Method next	290
Method offsetExists	291
Method offsetGet	291
Method offsetSet	292

Method offsetUnset	293
Method rewind	293
Method valid	294
Method get	294
Method set state	295
Method wakeup	295
Class OTS Player	295
Method ban	297
Method delete	298
Method find	298
Method getAccount	299
Method getCap	299
Method getConditions	299
Method getCustomField	300
Method getDepot	301
Method getDirection	301
Method getExperience	302
Method getGroup	302
Method getGuildNick	302
Method getHealth	303
Method getHealthMax	303
Method getId	304
Method getLastIP	304
Method getLastLogin	304
Method getLevel	305
Method getLookAddons	305
Method getLookBody	305
Method getLookFeet	306
Method getLookHead	306
Method getLookLegs	307
Method getLookType	307
Method getLossExperience	307
Method getLossMana	308
Method getLossSkills	308
Method getMagLevel	309
Method getMana	309
Method getManaMax	309
Method getManaSpent	310
Method getName	310
Method getPosX	311
Method getPosY	311
Method getPosZ	311
Method getPremiumEnd	312
Method getRank	312
Method getRankId	313
Method getRedSkullTime	313
Method getSave	313
Method getSex	314
Method getSkill	314

Method getSkillTries	315
Method getSlot	315
Method getSoul	316
Method getTownId	316
Method getVocation	317
Method getVocationName	317
Method hasRedSkull	318
Method isBanned	318
Method isLoaded	318
Method isSaveSet	319
Method load	319
Method save	320
Method setAccount	320
Method setCap	320
Method setConditions	321
Method setCustomField	321
Method setDepot	322
Method setDirection	323
Method setExperience	323
Method setGroup	324
Method setGuildNick	324
Method setHealth	325
Method setHealthMax	325
Method setLastIP	326
Method setLastLogin	326
Method setLevel	326
Method setLookAddons	327
Method setLookBody	327
Method setLookFeet	328
Method setLookHead	328
Method setLookLegs	329
Method setLookType	329
Method setLossExperience	330
Method setLossMana	330
Method setLossSkills	331
Method setMagLevel	331
Method setMana	331
Method setManaMax	332
Method setManaSpent	332
Method setName	333
Method setPosX	333
Method setPosY	334
Method setPosZ	334
Method setPremiumEnd	335
Method setRank	335
Method setRankId	336
Method setRedSkull	336
Method setRedSkullTime	336
Method setSave	337

Method setSex	337
Method setSkill	338
Method setSkillTries	338
Method setSlot	339
Method setSoul	340
Method setTownId	340
Method setVocation	340
Method unban	341
Method unsetRedSkull	341
Method unsetSave	342
Method __get	342
Method __set	343
Method __sleep	343
Class OTS Players List	344
Method deletePlayer	344
Method init	344
Class OTS Spell	345
Constructor __construct	346
Method createConjure	346
Method getCharges	347
Method getConjure	347
Method getConjureCount	348
Method getConjureId	348
Method getID	348
Method getLevel	349
Method getMagicLevel	349
Method getMana	349
Method getName	350
Method getReagent	350
Method getReagentId	350
Method getSoul	351
Method getType	351
Method getVocations	351
Method getWords	352
Method hasParams	352
Method isAggressive	352
Method isAggressive	353
Method isEnabled	353
Method isFarUseAllowed	353
Method isLearnNeeded	354
Method isPremium	354
Method __get	354
Class OTS SpellsList	355
Class Constant SPELL_CONJURE	355
Class Constant SPELL_INSTANT	356
Class Constant SPELL_RUNE	356
Constructor __construct	356
Method getConjure	357
Method getConjuresList	357

Method <code>getInstant</code>	358
Method <code>getInstantsList</code>	358
Method <code>getRune</code>	358
Method <code>getRunesList</code>	359
Method <code>get</code>	359
Method <code>set state</code>	360
Class <code>OTS_SQLField</code>	360
 Constructor <code>construct</code>	361
 Method <code>getName</code>	361
 Method <code>getTable</code>	362
 Method <code>get</code>	362
Class <code>OTS_SQLFilter</code>	363
 Class Constant <code>CRITERIUM_AND</code>	363
 Class Constant <code>CRITERIUM_OR</code>	363
 Class Constant <code>OPERATOR_EQUAL</code>	364
 Class Constant <code>OPERATOR_GREATER</code>	364
 Class Constant <code>OPERATOR LIKE</code>	364
 Class Constant <code>OPERATOR_LOWER</code>	365
 Class Constant <code>OPERATOR_NEQUAL</code>	365
 Class Constant <code>OPERATOR_NGREATER</code>	365
 Class Constant <code>OPERATOR_NLIKE</code>	366
 Class Constant <code>OPERATOR_NLOWER</code>	366
 Method <code>addFilter</code>	366
 Method <code>compareField</code>	367
 Method <code>getTables</code>	368
 Method <code>get</code>	368
 Method <code>sleep</code>	368
 Method <code>toString</code>	369
Class <code>OTS_VocationsList</code>	369
 Constructor <code>construct</code>	370
 Method <code>count</code>	370
 Method <code>getIterator</code>	370
 Method <code>getVocationId</code>	371
 Method <code>getVocationName</code>	371
 Method <code>offsetExists</code>	372
 Method <code>offsetGet</code>	372
 Method <code>offsetSet</code>	373
 Method <code>offsetUnset</code>	373
 Method <code>set state</code>	374
Class <code>POT</code>	374
 Class Constant <code>BAN_ACCOUNT</code>	375
 Class Constant <code>BAN_IP</code>	375
 Class Constant <code>BAN_PLAYER</code>	375
 Class Constant <code>DB_MYSQL</code>	376
 Class Constant <code>DB_ODBC</code>	376
 Class Constant <code>DB_PGSQL</code>	377
 Class Constant <code>DB_SQLITE</code>	377
 Class Constant <code>DEPOT_SID_FIRST</code>	377
 Class Constant <code>DIRECTION_EAST</code>	378

Class Constant DIRECTION NORTH	378
Class Constant DIRECTION SOUTH	378
Class Constant DIRECTION WEST	379
Class Constant ORDER ASC	379
Class Constant ORDER DESC	379
Class Constant SEX FEMALE	380
Class Constant SEX MALE	380
Class Constant SKILL AXE	381
Class Constant SKILL CLUB	381
Class Constant SKILL DISTANCE	381
Class Constant SKILL FISHING	382
Class Constant SKILL FIST	382
Class Constant SKILL SHIELDING	383
Class Constant SKILL SWORD	383
Class Constant SLOT AMMO	383
Class Constant SLOT ARMOR	384
Class Constant SLOT BACKPACK	384
Class Constant SLOT FEET	385
Class Constant SLOT HEAD	385
Class Constant SLOT LEFT	385
Class Constant SLOT LEGS	386
Class Constant SLOT NECKLACE	386
Class Constant SLOT RIGHT	387
Class Constant SLOT RING	387
Class Constant SPELL CONJURE	388
Class Constant SPELL INSTANT	388
Class Constant SPELL RUNE	388
Class Constant VOCATION DRUID	389
Class Constant VOCATION KNIGHT	389
Class Constant VOCATION NONE	390
Class Constant VOCATION PALADIN	390
Class Constant VOCATION SORCERER	390
Method areHousesLoaded	391
Method areItemsLoaded	391
Method areMonstersLoaded	391
Method areSpellsLoaded	392
Method areVocationsLoaded	392
Method banIP	393
Method connect	393
example: connect.php	393
Method createFilter	395
Method createObject	395
Method getConjure	395
Method getConjuresList	396
Method getDBHandle	397
Method getHousesList	397
Method getInstance	398
Method getInstant	398
Method getInstantsList	398

Method getItemList	399
Method getMap	399
Method getMonster	400
Method getMonstersList	400
Method getRune	401
Method getRunesList	401
Method getSpellsList	402
Method getVocationId	402
Method getVocationName	403
Method getVocationsList	403
Method isIPBanned	404
Method isMapLoaded	404
Method loadClass	405
Method loadHouses	405
Method loadItems	406
Method loadMap	406
Method loadMonsters	407
Method loadSpells	407
Method loadVocations	408
Method serverStatus	409
example: example	409
Method setItemsCache	410
Method setMapCache	410
Method setPOTPath	411
example: fakeroot.php	411
Method unbanIP	411
Method unloadHouses	412
Method unloadItems	412
Method unloadMap	413
Method unloadMonsters	413
Method unloadSpells	414
Method unloadVocations	414
compat.php	415
Appendices	416
Appendix A - Class Trees	417
POT	417
Appendix B - README/CHANGELOG/INSTALL	423
LICENSE	424
CHANGELOG	426
RULES	428
README	429
INSTALL	431
NEWS	431
Appendix D - Todo List	433

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 for 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 developement 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** 0.0.6
- **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_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+SVN
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.1.0+SVN
- **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+SVN
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.1.0+SVN
- **License** [GNU Lesser General Public License, Version 3](#)

IOTS_DAO.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0+SVN
- **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_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** 0.0.1
- **Version** 0.1.0+SVN
- **Copyright** 2007 (C) by Wrzasq
- **TODO** 0.2.0: Implement NetworkMessage.
- **TODO** 0.2.0: Implement OutOfBoundsException instead of mixed results types.
- **TODO** 1.0.0: Create automatic wrappers for related OTS resource handlers.
- **TODO** 0.1.1: Support for call constructors with ID/name parameter for automatic pre-load for data.
- **TODO** 0.1.0: Implement __toString() as display drivers.
- **TODO** 1.0.0: Unify *List and *_List naming (probably into *List).
- **TODO** 0.1.2: OTAdmin protocol.
- **TODO** 0.1.3: SOAP interface for remote controll.
- **TODO** 1.0.0: More detailed documentation and tutorials, also update examples and tutorials.
- **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: Deprecations cleanup.
- **TODO** 1.0.0: PHAR and PHK packages.
- **TODO** 1.0.0: Main POT class as database instance.
- **TODO** 1.0.0: Complete phpUnit test.
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Account.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Copyright** 2007 (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** 0.0.5
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Base_DAO.php

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

OTS_Container.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0+SVN
- **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+SVN
- **Version** 0.0.6
- **Copyright** 2007 (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** 0.1.0+SVN
- **Version** 0.0.6
- **Copyright** 2007 (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** 0.1.0+SVN
- **Version** 0.0.1
- **Copyright** 2007 (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** 0.0.5
- **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** 0.1.0+SVN
- **Version** 0.0.4
- **Copyright** 2007 (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** 0.1.0+SVN
- **Version** 0.0.4
- **Copyright** 2007 (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** 0.0.5
- **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** 0.0.5
- **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+SVN
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.1.0+SVN
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_HousesList.php

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

OTS_InfoRespond.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0+SVN
- **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+SVN
- **Version** 0.0.3
- **Copyright** 2007 (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+SVN
- **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+SVN
- **Version** 0.0.8
- **Copyright** 2007 (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+SVN
- **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+SVN
- **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+SVN
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.1.0+SVN
- **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+SVN
- **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** 0.1.0+SVN
- **Version** 0.0.1
- **Copyright** 2007 (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** 0.0.5
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Spell.php

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

OTS_SQLField.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0+SVN
- **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+SVN
- **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_VocationsList.php

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

Package POT Classes

Class E_OTS_ErrorCode

[line 20]

Generic exception class for error codes.

Generic exception class for error codes.

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

Constructor *void* function E_OTS_ErrorCode::__construct(\$code) [line 27]

Function Parameters:

- *int* **\$code** Error code.

Sets error code.

Sets error code.

- **Version** 0.0.6
- **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_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+SVN
- **Since** 0.1.0+SVN

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_OTBMErrors

[line 22]

OTBM map loading error.

OTBM map loading error.

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

E_OTS_OTBMErrors::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+SVN
- **Since** 0.1.0+SVN

Class IOTS_DAO

[line 23]

OTserv database object.

OTserv database object.

This insterface indicates that class is a OTServ DAO class.

- **Package** POT
- **Version** 0.1.0+SVN
- **Deprecated** 0.1.0+SVN 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_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 29]

OTServ account abstraction.

OTServ account abstraction.

- **Package** POT
- **Property** string \$password: Password.
- **Property** string \$eMail: Email address.
- **Property-read** int \$id: Account number.
- **Property-read** bool \$isLoading: Loaded state.
- **Property-read** [OTS Players List](#) \$playersList: Characters of this account.
- **Property-read** bool \$isBlocked: Blocked flag state.
- **Property-read** bool \$isBanned: Ban state.
- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Since** 0.0.1

void function OTS_Account::ban([\$time = 0]) [line 471]

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 317*]

Blocks account.

Blocks account.

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

int function OTS_Account::count() [*line 562*]

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 54]

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** 0.0.6
- **Version** 0.0.1
- **Throws** Exception 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 122*]

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_SVN
- **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 525*]

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 146*]

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

string function OTS_Account::getCustomField(\$field) [*line 364*]

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 269*]

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 211*]

Returns group of this account.

Returns group of this account.

- **Version** 0.1.0+SVN
- **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::getId()* [*line 192*]

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 549*]

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 330*]

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 242*]

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 414*]

List of characters on account.

List of characters on account.

- **Version** 0.1.0+SVN
- **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 444*]

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+SVN
- **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 506*]

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 296*]

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 163*]

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 133]

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 174]

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 390]

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 284]

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 231]

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 347]

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 257*]

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 488*]

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::unlock() [*line 309*]

Unblocks account.

Unblocks account.

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

mixed function OTS_Account::__get(\$name) [*line 576*]

Function Parameters:

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

Magic PHP5 method.

Magic PHP5 method.

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

void function OTS_Account::__set(\$name, \$value) [*line 615*]

Function Parameters:

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

Magic PHP5 method.

Magic PHP5 method.

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

Class OTS_Accounts_List

[*line 21*]

List of accounts.

List of accounts.

- **Package** POT
- **Version** 0.0.5

- **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

Class OTS_Base_DAO

[line 22]

Basic data access object routines.

Basic data access object routines.

- **Package** POT
- **Version** 0.1.0+SVN
- **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+SVN
- **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+SVN
- **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+SVN
- **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($db)` [*line 88*]

Function Parameters:

- **PDO \$db** Database connection object.

Sets database connection handler.

Sets database connection handler.

- **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

OTS_DAO function `OTS_Base_List::current()` [*line 193*]

Returns current row.

Returns current row.

- **Version** 0.1.0+SVN
- **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+SVN
- **Version** 0.0.5
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN
- **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_Container

[line 22]

Container item representation.

Container item representation.

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

- **Since** 0.1.0+SVN
- **Access** public

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

Current cursor position.

Current cursor position.

- **Version** 0.0.3
- **Deprecated** 0.1.0+SVN 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+SVN 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 it's 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+SVN 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+SVN 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 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_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 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_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 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_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 IOTS_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** [LOTS_FileCache](#) \$cacheDriver: Cache driver.
- **Version** 0.1.0+SVN
- **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 130]

Function Parameters:

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

Opens file.

Opens file.

- **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 328]

Function Parameters:

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

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0+SVN
- **Version** 0.0.6
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN
- **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 33]

OTServ binary file node representation.

OTServ binary file node representation.

- **Package** POT
- **Property** string \$buffer: Properties binary string.
- **Property** [OTS_FileNode](#) \$next: Next sibling node.
- **Property** [OTS_FileNode](#) \$child: First child node.
- **Property** int \$type: Node type.
- **Property-read** bool \$isValid: isValid() method wrapper.
- **Property-read** int \$char: getChar() method wrapper.
- **Property-read** int \$short: getShort() method wrapper.
- **Property-read** int \$long: getLong() method wrapper.
- **Property-read** string \$string: getString(false) call wrapper.
- **Version** 0.1.0+SVN
- **Version** 0.0.6
- **Since** 0.0.6

string function OTS_FileNode::getBuffer() [*line 113*]

Returns properties stream.

Returns properties stream.

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

int function OTS_FileNode::getChar() [*line 218*]

Returns single byte.

Returns single byte.

- **Version** 0.0.6

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

OTS_FileNode function *OTS_FileNode::getChild()* [*line 154*]

Returns first child.

Returns first child.

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

int function *OTS_FileNode::getLong()* [*line 248*]

Returns quater byte.

Returns quater byte.

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

OTS_FileNode function *OTS_FileNode::getNext()* [*line 134*]

Returns next sibling.

Returns next sibling.

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

- **Access** public

int function OTS_FileNode::getShort() [*line 233*]

Returns double byte.

Returns double byte.

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

string function OTS_FileNode::getString([\$length = false]) [*line 266*]

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

int function OTS_FileNode::getType() [*line 174*]

Returns node type.

Returns node type.

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

bool function OTS_FileNode::isValid() [*line 194*]

Checks if there is anything left in stream.

Checks if there is anything left in stream.

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

void function OTS_FileNode::setBuffer(\$buffer) [*line 123*]

Function Parameters:

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

Sets properties stream.

Sets properties stream.

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

void function OTS_FileNode::setChild(\$child) [line 164]

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 144]

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 184]

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::skip(\$n) [line 288]

Function Parameters:

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

Skips given amount of bytes.

Skips given amount of bytes.

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

void function OTS_FileNode::__clone() [line 73]

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 303*]

Function Parameters:

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

Magic PHP5 method.

Magic PHP5 method.

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

void function OTS_FileNode::__set(\$name, \$value) [*line 348*]

Function Parameters:

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

Magic PHP5 method.

Magic PHP5 method.

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

`void function OTS_FileNode::__set_state($properties) [line 95]`
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_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 \$isLoaded: Loaded state check.
- **Property-read** int \$id: Row ID.
- **Property-read** [OTS_Players_List](#) \$playersList: List of members of this group.
- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Since** 0.0.1

int function OTS_Group::count() [*line 393*]

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 356*]

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

int function OTS_Group::getAccess() [*line 162*]

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 249*]

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 135*]

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 91*]

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 380*]

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 189*]

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 216*]

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 108*]

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 299*]

List of characters in given group.

List of characters in given group.

- **Version** 0.1.0+SVN
- **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 329*]

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+SVN
- **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::isLoading()` [*line 56*]

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 66]

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 177]

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 275*]

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 150]

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 204]

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 231*]

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 123*]

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 407*]

Function Parameters:

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

Magic PHP5 method.

Magic PHP5 method.

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

void function OTS_Group::__set(\$name, \$value) [*line 449*]

Function Parameters:

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

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN

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

Class OTS_Groups_List

[line 21]

List of groups.

List of groups.

- **Package** POT
- **Version** 0.0.5
- **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

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** [OTS_GuildAction](#) \$invitesDriver: Invitations handler.
- **Property-write** [OTS_GuildAction](#) \$requestsDriver: Membership requests handler.
- **Version** 0.1.0+SVN
- **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_OTs_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_OTS_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+SVN
- **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+SVN
- **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+SVN
- **Version** 0.0.4
- **Throws** E_OTS_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_OTS_NotLoaded If guild is not loaded.
- **Throws** E_OTS_NoDriver If there is no invites driver assigned.
- **Since** 0.0.4
- **Access** public

bool function OTS_Guild::isLoaded() [*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_OTTS_NotLoaded If guild is not loaded.
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::setInvitesDriver([\$invites = null]) [line 90]

Function Parameters:

- [*IOOTS_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+SVN
- **Version** 0.0.4
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN
- **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+SVN
- **Version** 0.0.4
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN
- **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

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 \$isLoading: Loaded state check.
- **Property-read** int \$id: Row ID.
- **Property-read** [OTS_Players_List](#) \$playersList: List of members with this rank.
- **Version** 0.1.0+SVN
- **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+SVN
- **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+SVN
- **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+SVN
- **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_OTTS_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+SVN
- **Version** 0.0.4

- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN
- **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+SVN
- **Version** 0.0.4
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN
- **Since** 0.0.4
- **Access** public

Class OTS_GuildRanks_List

[*line 21*]

List of guild ranks.

List of guild ranks.

- **Package** POT
- **Version** 0.0.5
- **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

Class OTS_Guilds_List

[line 21]

List of guilds.
List of guilds.

- **Package** POT
- **Version** 0.0.5
- **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

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** string \$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+SVN
- **Since** 0.1.0+SVN

Constructor *void* function OTS_House::__construct(\$element) [*line 52*]

Function Parameters:

- *DOMElement* **\$element** House information.

Creates wrapper for given house element.
Creates wrapper for given house element.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

void function OTS_House::addTile(\$tile) [*line 264*]

Function Parameters:

- [OTS_MapCoords](#) **\$tile** Tile to be added.

Adds tile to house.
Adds tile to house.

- **Version** 0.1.0+SVN

- **Since** 0.1.0+SVN
- **Access** public

void function OTS_House::delete() [*line 94*]

Deletes house info from database.

Deletes house info from database.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

OTS_MapCoords function OTS_House::getEntry() [*line 168*]

Returns entry position.

Returns entry position.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

int function OTS_House::getId() [*line 108*]

Returns house's ID.

Returns house's ID.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN

- **Access** public

string function OTS_House::getName() [*line 118*]

Return house's name.

Return house's name.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

OTS_Player|null function OTS_House::getOwner() [*line 178*]

Returns current house owner.

Returns current house owner.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

int|false function OTS_House::getPaid() [*line 208*]

Returns paid date.

Returns paid date.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

int function OTS_House::getRent() [*line 148*]

Returns house rent cost.

Returns house rent cost.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

int function OTS_House::getSize() [*line 158*]

Returns house size.

Returns house size.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

array function OTS_House::getTiles() [*line 274*]

Returns tiles list.

Returns tiles list.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

int function OTS_House::getTownId() [*line 128*]

Returns town ID in which house is located.

Returns town ID in which house is located.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

string function OTS_House::getTownName() [*line 138*]

Returns town name.

Returns town name.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

string/false function OTS_House::getWarnings() [*line 236*]

Returns house warnings.

Returns house warnings.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

void function OTS_House::save() [*line 77*]

Saves info in database.

Saves info in database.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

void function OTS_House::setOwner(\$player) [line 198]

Function Parameters:

- [*OTS_Player*](#) **\$player** House owner to be set.

Sets house owner.

Sets house owner.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

void function OTS_House::setPaid(\$paid) [line 226]

Function Parameters:

- *int* **\$paid** Sets paid timestamp to passed one.

Sets paid date.

Sets paid date.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

void function OTS_House::setWarnings(\$warnings) [*line 254*]

Function Parameters:

- *string* **\$warnings** Sets house warnings.

Sets house warnings.

Sets house warnings.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

mixed function OTS_House::__get(\$name) [*line 286*]

Function Parameters:

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

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0+SVN
- **Throws** OutOfBoundsException For non-supported properties.

- **Since** 0.1.0+SVN
- **Access** public

void function OTS_House::__set(\$name, \$value) [*line 335*]

Function Parameters:

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

Magic PHP5 method.

Magic PHP5 method.

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

array function OTS_House::__sleep() [*line 69*]

Magic PHP5 method.

Magic PHP5 method.

Allows object serialisation.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

Class OTS_HousesList

[line 20]

Wrapper for houses list.

Wrapper for houses list.

- **Package** POT
- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN

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+SVN
- **Since** 0.1.0+SVN
- **Access** public

int function OTS_HousesList::count() [line 109]

Returns amount of houses.

Returns amount of houses.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **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+SVN
- **Since** 0.1.0+SVN
- **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+SVN

- **Since** 0.1.0+SVN
- **Access** public

ArrayIterator function `OTS_HousesList::getIterator()` [*line 119*]

Returns iterator handle for loops.

Returns iterator handle for loops.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **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+SVN
- **Since** 0.1.0+SVN
- **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+SVN
- **Since** 0.1.0+SVN
- **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+SVN
- **Throws** E_OTS_ReadOnly Always - this class is read-only.
- **Since** 0.1.0+SVN
- **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+SVN
- **Throws** E_OTS_ReadOnly Always - this class is read-only.
- **Since** 0.1.0+SVN
- **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+SVN
- **Since** 0.1.0+SVN
- **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+SVN

- **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+SVN
- **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+SVN
- **Version** 0.0.2
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN
- **Since** 0.0.2
- **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+SVN

- **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+SVN
- **Version** 0.0.3
- **Since** 0.0.3
- **Since** 0.1.0+SVN
- **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+SVN
- **Version** 0.0.3
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN
- **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+SVN
- **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 480*]

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 491*]

Returns item at current position in iterator.

Returns item at current position in iterator.

- **Version** 0.0.8
- **Deprecated** 0.1.0+SVN Use getIterator().
- **Since** 0.0.8
- **Access** public

int function OTS_ItemsList::getBuildVersion() [*line 416*]

Returns build version.

Returns build version.

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

int function OTS_ItemsList::getClientVersion() [*line 406*]

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 427*]

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 449*]

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 than 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 470*]

Returns all loaded items.

Returns all loaded items.

- **Version** 0.0.8
- **Deprecated** 0.1.0+SVN 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 545*]

Returns iterator handle for loops.

Returns iterator handle for loops.

- **Version** 0.1.0+SVN
- **Version** 0.0.8
- **Since** 0.0.8

- **Since** 0.1.0+SVN
- **Access** public

int function OTS_ItemsList::getOTBVersion() [*line 396*]

Returns OTB file version.

Returns OTB file version.

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

int function OTS_ItemsList::key() [*line 512*]

Returns ID of current position.

Returns ID of current position.

- **Version** 0.0.8
- **Deprecated** 0.1.0+SVN 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 501*]

Moves to next iterator item.

Moves to next iterator item.

- **Version** 0.0.8
- **Deprecated** 0.1.0+SVN Use getIterator().
- **Since** 0.0.8
- **Access** public

bool function OTS_ItemsList::offsetExists(\$offset) [*line 558*]

Function Parameters:

- *string|int* **\$offset** Array key.

Checks if given element exists.

Checks if given element exists.

- **Version** 0.1.0+SVN
- **Version** 0.0.8
- **Since** 0.0.8

- **Since** 0.1.0+SVN
- **Access** public

mixed function OTS_ItemsList::offsetGet(\$offset) [*line 580*]

Function Parameters:

- *string|int* **\$offset** Array key.

Returns item from given position.

Returns item from given position.

- **Version** 0.1.0+SVN
- **Version** 0.0.8
- **Since** 0.0.8
- **Since** 0.1.0+SVN
- **Access** public

void function OTS_ItemsList::offsetSet(\$offset, \$value) [*line 611*]

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+SVN
- **Version** 0.0.8
- **Throws** E_OTS_ReadOnly Always - this class is read-only.
- **Since** 0.1.0+SVN
- **Since** 0.0.8
- **Access** public

void function OTS_ItemsList::offsetUnset(\$offset) [line 624]
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+SVN
- **Version** 0.0.8
- **Throws** E_OTS_ReadOnly Always - this class is read-only.
- **Since** 0.1.0+SVN
- **Since** 0.0.8
- **Access** public

void function OTS_ItemsList::rewind() [line 533]
Resets iterator index.
 Resets iterator index.

- **Version** 0.0.8
- **Deprecated** 0.1.0+SVN Use getIterator().
- **Since** 0.0.8
- **Access** public

bool function OTS_ItemsList::valid() [*line 523*]

Checks if there is anything more in interator.

Checks if there is anything more in interator.

- **Version** 0.0.8
- **Deprecated** 0.1.0+SVN Use getIterator().
- **Since** 0.0.8
- **Access** public

mixed function OTS_ItemsList::__get(\$name) [*line 638*]

Function Parameters:

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

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0+SVN
- **Version** 0.0.8
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN

- **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 \$isBlocking: Is item blocking move.
- **Property-read** bool \$hasHeight: Does item have height.
- **Property-read** bool \$isUsable: Is item usable.
- **Property-read** bool \$isPickupable: Is player able to pick it up.
- **Property-read** bool \$isMovable: Can be moved.
- **Property-read** bool \$isStackable: Can be stacked.
- **Property-read** bool \$isAlwaysOnTop: Is always on top of stack.
- **Property-read** bool \$isReadable: Has readable sign.
- **Property-read** bool \$isRotable: Can be rotated.
- **Property-read** bool \$isHangable: Can be hang.
- **Property-read** bool \$isVertical: Is vertically oriented.
- **Property-read** bool \$isHorizontal: Is horizontally oriented.
- **Property-write** int \$flags: Special flags.
- **Version** 0.1.0+SVN
- **Version** 0.0.8

- **Since** 0.0.8

OTS_ItemType::FLAG_ALLOWDISTREAD

= 1048576 [*line 221*]

Can be read from distance.

Can be read from distance.

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

OTS_ItemType::FLAG_ALWAYSONTOP

= 8192 [*line 193*]

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 149*]

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 145*]

BLOCK_PROJECTILE flag(?).
BLOCK_PROJECTILE flag(?).

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

OTS_ItemType::FLAG_BLOCK_SOLID

= 1 [*line 141*]

Can block characters from walking.
Can block characters from walking.

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

OTS_ItemType::FLAG_CANNOTDECAY

= 524288 [*line 217*]

Doesn't decay.
Doesn't decay.

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

OTS_ItemType::FLAG_FLOORCHANGEDOWN

= 256 [*line 173*]

Changes floor under it.

Changes floor under it.

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

OTS_ItemType::FLAG_FLOORCHANGE EAST

= 1024 [*line 181*]

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 177*]

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 185*]

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 189*]

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 205*]

Can be hang(?).
Can be hang(?).

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

OTS_ItemType::FLAG_HAS_HEIGHT

= 8 [*line 153*]

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 213*]

Is oriented horizontally.
Is oriented horizontally.

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

OTS_ItemType::FLAG_MOVEABLE

= 64 [*line 165*]

Can be moved by player.
Can be moved by player.

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

OTS_ItemType::FLAG_PICKUPABLE

= 32 *[line 161]*

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 197]*

Has readable sign.
Has readable sign.

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

OTS_ItemType::FLAG_ROTABLE

= 32768 *[line 201]*

Can be rotated by player.
Can be rotated by player.

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

OTS_ItemType::FLAG_STACKABLE

= 128 [*line 169*]

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 157*]

Can be used by players.
Can be used by players.

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

OTS_ItemType::FLAG_VERTICAL

= 131072 [*line 209*]

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_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 121*]

Container.
Container.

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

OTS_ItemType::ITEM_TYPE_DEPOT

= 1 [*line 109*]

Depot locker.
Depot locker.

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

OTS_ItemType::ITEM_TYPE_DOOR

= 5 [*line 125*]

Door.
Door.

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

OTS_ItemType::ITEM_TYPE_MAGICFIELD

= 6 [*line 129*]

Magic field.

Magic field.

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

OTS_ItemType::ITEM_TYPE_MAILBOX

= 2 [*line 113*]

Mailbox.

Mailbox.

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

OTS_ItemType::ITEM_TYPE_NONE

= 0 [*line 105*]

No special type.

No special type.

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

OTS_ItemType::ITEM_TYPE_TELEPORT

= 7 [*line 136*]

Teleport.

Teleport.

- **Version** 0.1.0+SVN
- **Version** 0.0.8
- **Since** 0.0.8
- **Since** 0.1.0+SVN

OTS_ItemType::ITEM_TYPE_TRASHHOLDER

= 3 [*line 117*]

Trash can.

Trash can.

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

Constructor *void* function OTS_ItemType::__construct(\$id) [*line 277*]

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 567*]

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, \$attribbyte) [*line 359*]

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 387*]

Returns all attributes list.

Returns all attributes list.

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

int function OTS_ItemType::getClientId() [*line 318*]

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 397*]

Returns group.

Returns group.

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

int function OTS_ItemType::getId() [*line 308*]

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 338*]

Returns item name.

Returns item name.

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

int function OTS_ItemType::getType() [*line 417*]

Returns item type.

Returns item type.

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

bool function OTS_ItemType::hasHeight() [*line 457*]

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 507*]

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 447*]

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 537*]

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 557*]

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 487*]

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 477*]

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 517*]

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 527*]

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 497*]

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 467*]

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 547*]

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 377*]

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 328*]

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 437*]

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 407]

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 348]

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 427*]

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 590*]

Function Parameters:

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

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0+SVN

- **Version** 0.0.8
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN
- **Since** 0.0.8
- **Access** public

void function OTS_ItemType::__set(\$name, \$value) [*line 662*]

Function Parameters:

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

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0+SVN
- **Version** 0.0.8
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN
- **Since** 0.0.8
- **Access** public

void function OTS_ItemType::__set_state(\$properties) [*line 290*]

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+SVN
- **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+SVN
- **Version** 0.0.6
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN
- **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+SVN
- **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+SVN
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.1.0+SVN
- **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+SVN Use getItems().
- **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+SVN
- **Version** 0.0.6
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN
- **Since** 0.0.6
- **Access** public

Class OTS_MonstersList

[line 20]

Wrapper for monsters list.
Wrapper for monsters list.

- **Package** POT
- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN

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+SVN
- **Since** 0.1.0+SVN
- **Access** public

int function OTS_MonstersList::count() [line 110]
Returns amount of monsters loaded.
Returns amount of monsters loaded.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN

- **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+SVN
- **Since** 0.1.0+SVN
- **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+SVN
- **Since** 0.1.0+SVN
- **Access** public

string function OTS_MonstersList::key() [*line 138*]

Returns name of current position.

Returns name of current position.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

void function OTS_MonstersList::next() [*line 128*]

Moves to next iterator monster.

Moves to next iterator monster.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **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+SVN
- **Since** 0.1.0+SVN
- **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+SVN
- **Since** 0.1.0+SVN
- **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+SVN
- **Throws** E_OTS_ReadOnly Always - this class is read-only.
- **Since** 0.1.0+SVN
- **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+SVN
- **Throws** E_OTS_ReadOnly Always - this class is read-only.
- **Since** 0.1.0+SVN
- **Access** public

void function OTS_MonstersList::rewind() [*line 156*]

Resets iterator index.

Resets iterator index.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

bool function OTS_MonstersList::valid() [*line 148*]

Checks if there is anything more in iterator.

Checks if there is anything more in iterator.

- **Version** 0.1.0+SVN

- **Since** 0.1.0+SVN
- **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+SVN
- **Since** 0.1.0+SVN
- **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+SVN
- **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_HOusetile

= 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+SVN 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+SVN
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.1.0+SVN
- **Access** public

ArrayIterator function OTS_OTBMFile::getIterator() [*line 557*]

Returns iterator handle for loops.

Returns iterator handle for loops.

- **Version** 0.1.0+SVN
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.1.0+SVN
- **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+SVN 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+SVN 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+SVN
- **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+SVN 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+SVN
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.1.0+SVN
- **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+SVN
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.1.0+SVN
- **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+SVN
- **Version** 0.0.6
- **Throws** E_OTS_ReadOnly Always - this class is read-only.
- **Since** 0.1.0+SVN
- **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+SVN
- **Version** 0.0.6
- **Throws** E_OTS_ReadOnly Always - this class is read-only.
- **Since** 0.1.0+SVN
- **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+SVN 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+SVN 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+SVN
- **Version** 0.0.6
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN
- **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 62]

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-read** int \$id: Player ID.
- **Property-read** bool \$isSaveSet: Player save flag.
- **Property-read** bool \$hasRedSkull: Player red skull flag.
- **Property-read** bool \$isBanned: Player banned state.
- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Since** 0.0.1

void function OTS_Player::ban([\$time = 0]) [*line 1719*]

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 1773]

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 124]

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 226]

Returns account of this player.

Returns account of this player.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Throws** *E_OTS_NotLoaded* If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function *OTS_Player::getCap()* [line 881]

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 1019]

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 1319*]

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 1598*]

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** 0.1.0+SVN
- **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 611*]

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 368*]

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 255*]

Returns group of this player.

Returns group of this player.

- **Version** 0.1.0+SVN
- **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 1106*]

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 449*]

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 476*]

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

int function OTS_Player::getId() [*line 182*]

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 935*]

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 908*]

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 395*]

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 773*]

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 638*]

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 665*]

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 692*]

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 719*]

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 746*]

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 1230*]

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 1257*]

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 1284*]

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 422*]

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 503*]

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 530*]

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 557*]

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 199*]

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 800*]

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 827*]

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 854*]

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 285*]

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 1151*]

Assigned guild rank.

Assigned guild rank.

- **Version** 0.1.0+SVN
- **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 1134*]

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 1046*]

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 991*]

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 314*]

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 1374*]

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 1406*]
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 1460*]
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** 0.1.0+SVN
- **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 584*]

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 function OTS_Player::getTownId() [*line 1203*]

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

int function OTS_Player::getVocation() [*line 341*]

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 1796*]

Player profession name.

Player proffesion name.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Deprecated** 0.1.0+SVN Use POT::getInstance()->getVocationsList().
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1

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

bool function OTS_Player::hasRedSkull() [line 1073]

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 1754]

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 141]

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 962*]

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 102*]

Function Parameters:

- *int* **\$id** Player's ID.

Loads player with given id.

Loads player with given id.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1

- **Access** public

void function OTS_Player::save() [line 151]

Saves player in database.

Saves player in database.

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

void function OTS_Player::setAccount(\$account) [line 243]

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::setCap(\$cap) [line 896]

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 1034]

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 1349]

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], [\$spid = 0], [\$depot_id = 0]) [line 1655]

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* **\$spid** Deprecated, not used anymore.
- *int* **\$depot_id** Internal, for further use.

Sets depot content.

Sets depot content.

- **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::setDirection(\$direction) [line 626]

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 383]

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 272*]

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 1121*]

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 464*]
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 491*]
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 950]

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 923]

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 410]

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 788*]

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 653*]

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 680]

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 707]

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 734]

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 761]

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 1245]

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 1272]

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 1299]

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 437]

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 518]

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 545]

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 572]

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 214]

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 815]

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 842]

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 869]

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 302]

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 1184]

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 1174]

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 1094]

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 1061]

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 1007]

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 329]

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 1392]

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 1424]

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 1516]

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** 0.0.5
- **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::setSoul(\$soul) [line 599]

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::setTownId(\$town_id) [line 1218]

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 356]

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 1736]

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 1086]

Unsets red skull flag.

Unsets red skull flag.

- **Version** 0.0.1
- **Since** 0.0.1

- **Access** public

void function OTS_Player::unsetSave() [*line 977*]

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 1815*]

Function Parameters:

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

Magic PHP5 method.

Magic PHP5 method.

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

void function OTS_Player::__set(\$name, \$value) [line 1953]

Function Parameters:

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

Magic PHP5 method.

Magic PHP5 method.

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

array function OTS_Player::__sleep() [line 91]

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

Class OTS_Players_List

[line 21]

List of players.

List of players.

- **Package** POT
- **Version** 0.0.5
- **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

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 \$isAgressive: 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 \$isEnabled: Is spell enabled.
- **Property-read** bool \$isFarUseAllowed: Can the spell be used from distance.
- **Property-read** bool \$isPremium: Does spell requires PACC.
- **Property-read** bool \$isLearnNeeded: 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+SVN
- **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+SVN
- **Version** 0.0.7
- **Since** 0.0.7
- **Since** 0.1.0+SVN
- **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+SVN
- **Version** 0.0.7
- **Since** 0.0.7
- **Since** 0.1.0+SVN
- **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+SVN 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+SVN
- **Version** 0.0.7
- **Since** 0.0.7
- **Since** 0.1.0+SVN
- **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+SVN 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+SVN
- **Version** 0.0.7
- **Deprecated** 0.1.0+SVN 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+SVN
- **Version** 0.0.7
- **Since** 0.0.7
- **Since** 0.1.0+SVN
- **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+SVN
- **Version** 0.0.7
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN
- **Since** 0.0.7
- **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+SVN
- **Since** 0.1.0+SVN

OTS_SpellsList::SPELL_CONJURE

= 2 [line 36]

Conjure spell.

Conjure spell.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN

OTS_SpellsList::SPELL_INSTANT

= 1 [*line 32*]

Instant spell.

Instant spell.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN

OTS_SpellsList::SPELL_RUNE

= 0 [*line 28*]

Rune spell.

Rune spell.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN

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+SVN
- **Since** 0.1.0+SVN
- **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+SVN
- **Since** 0.1.0+SVN
- **Access** public

array function OTS_SpellsList::getConjuresList() [*line 170*]

Returns list of conjure spells.

Returns list of conjure spells.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **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+SVN
- **Since** 0.1.0+SVN
- **Access** public

array function OTS_SpellsList::getInstantsList() [*line 142*]

Returns list of instants.
Returns list of instants.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **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+SVN
- **Since** 0.1.0+SVN
- **Access** public

array function OTS_SpellsList::getRunesList() [*line 114*]

Returns list of runes.

Returns list of runes.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **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+SVN
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN
- **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+SVN
- **Since** 0.1.0+SVN
- **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+SVN
- **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+SVN
- **Version** 0.0.5
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN
- **Since** 0.0.5
- **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+SVN
- **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 241*]

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 254]

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 266*]

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 324*]

Function Parameters:

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

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0+SVN
- **Version** 0.0.5
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0+SVN
- **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 95*]

Returns string representation of WHERE clause.

Returns string representation of WHERE clause.

Returned string can be easily inserted into SQL query.

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

Class OTS_VocationsList

[*line 20*]

Wrapper for vocations.

Wrapper for vocations.xml file.

- **Package** POT
- **Version** 0.1.0+SVN

- **Since** 0.1.0+SVN

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+SVN
- **Since** 0.1.0+SVN
- **Access** public

int function OTS_VocationsList::count() [*line 104*]

Returns amount of vocations loaded.

Returns amount of vocations loaded.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **Access** public

ArrayIterator function OTS_VocationsList::getIterator() [*line 114*]

Returns iterator handle for loops.

Returns iterator handle for loops.

- **Version** 0.1.0+SVN
- **Since** 0.1.0+SVN
- **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+SVN
- **Since** 0.1.0+SVN
- **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+SVN

- **Since** 0.1.0+SVN
- **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+SVN
- **Since** 0.1.0+SVN
- **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+SVN
- **Since** 0.1.0+SVN
- **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+SVN
- **Throws** E_OTS_ReadOnly Always - this class is read-only.
- **Since** 0.1.0+SVN
- **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+SVN
- **Throws** E_OTS_ReadOnly Always - this class is read-only.
- **Since** 0.1.0+SVN
- **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+SVN
- **Since** 0.1.0+SVN
- **Access** public

Class POT

[line 38]

Main POT class.

Main POT class.

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

POT::BAN_ACCOUNT

= 3 [*line 269*]

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 255*]

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 262*]

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 43*]

MySQL driver.

MySQL driver.

- **Version** 0.0.1
- **Since** 0.0.1

POT::DB_ODBC

= 4 [*line 61*]

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 54]*

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 47]*

SQLite driver.
SQLite driver.

- **Version** 0.0.1
- **Since** 0.0.1

POT::DEPOT_SID_FIRST

= 100 *[line 247]*

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 110*]

East.

East.

- **Version** 0.0.1
- **Since** 0.0.1

POT::DIRECTION_NORTH

= 0 [*line 106*]

North.

North.

- **Version** 0.0.1
- **Since** 0.0.1

POT::DIRECTION_SOUTH

= 2 [*line 114*]

South.

South.

- **Version** 0.0.1
- **Since** 0.0.1

POT::DIRECTION_WEST

= 3 [*line 118*]

West.

West.

- **Version** 0.0.1
- **Since** 0.0.1

POT::ORDER_ASC

= 1 [*line 277*]

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 284*]

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 66*]

Female gender.
Female gender.

- **Version** 0.0.1
- **Since** 0.0.1

POT::SEX_MALE

= 1 [*line 70*]

Male gender.
Male gender.

- **Version** 0.0.1
- **Since** 0.0.1

POT::SKILL_AXE

= 3 [*line 147*]

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 133*]

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 154*]

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 168*]

Fishing.

Fishing.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2

POT::SKILL_FIST

= 0 [*line 126*]

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 161*]

Shielding.
Shielding.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2

POT::SKILL_SWORD

= 2 [*line 140*]

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 239*]

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 197*]

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 190*]

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 225*]

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 176*]

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 211*]

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 218*]

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 183*]

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 204*]

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 232*]

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 309]

Conjure spell.
Conjure spell.

- **Version** 0.0.7
- **Version** 0.0.1
- **Deprecated** 0.1.0+SVN Use OTS_SpellsList::SPELL_CONJURE.
- **Since** 0.0.7
- **Since** 0.0.1

POT::SPELL_INSTANT

= 1 [line 301]

Instant spell.
Instant spell.

- **Version** 0.0.7
- **Version** 0.0.1
- **Deprecated** 0.1.0+SVN Use OTS_SpellsList::SPELL_INSTANT.
- **Since** 0.0.7
- **Since** 0.0.1

POT::SPELL_RUNE

= 0 [line 293]

Rune spell.
Rune spell.

- **Version** 0.0.7
- **Version** 0.0.1
- **Deprecated** 0.1.0+SVN Use OTS_SpellsList::SPELL_RUNE.
- **Since** 0.0.7
- **Since** 0.0.1

POT::VOCATION_DRUID

= 2 [*line 89*]

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 101*]

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 77]*

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 95]*

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 83]*

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 1090*]

Checks if houses are loaded.

Checks if houses are loaded.

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

bool function POT::areItemsLoaded() [*line 1181*]

Checks if items are loaded.

Checks if items are loaded.

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

bool function POT::areMonstersLoaded() [*line 811*]

Checks if monsters are loaded.

Checks if monsters are loaded.

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

bool function POT::areSpellsLoaded() [*line 897*]

Checks if spells are loaded.

Checks if spells are loaded.

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

bool function POT::areVocationsLoaded() [*line 703*]

Checks if vocations are loaded.

Checks if vocations are loaded.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Since** 0.0.1

- **Since** 0.1.0+SVN
- **Access** public

`void function POT::banIP($ip, [$mask = '255.255.255.255'], [$time = 0])` [line 576]

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 432]

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** 0.0.4
- **Version** 0.0.1
- **Throws** Exception When driver is not supported.
- **Since** 0.0.1
- **Access** public
- **Example**

OTS_SQLFilter function POT::createFilter() [*line 669*]

Creates lists filter.

Creates lists filter.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Deprecated** 0.1.0+SVN Create objects directly from now.
- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

IOTS_DAO function POT::createObject(\$class) [*line 485*]

Function Parameters:

- *string* **\$class** Class name.

Creates OTServ DAO class instance.

Creates OTServ DAO class instance.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Deprecated** 0.1.0+SVN Create objects directly from now.
- **Since** 0.0.1
- **Access** public

OTS_Spell[*null*] function POT::getConjure(\$name) [*line 1050*]

Function Parameters:

- *string* **\$name** Spell name.

Returns given conjure spell.

Returns given conjure spell.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Deprecated** 0.1.0+SVN Use POT::getSpellsList()->getConjure().
- **Throws** E_OTS_NotLoaded If spells list is not loaded.
- **Since** 0.0.1
- **Since** 0.0.7
- **Access** public

array function POT::getConjuresList() [*line 1028*]

Returns list of conjure spells.

Returns list of conjure spells.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Deprecated** 0.1.0+SVN Use POT::getSpellsList()->getConjuresList().
- **Throws** E_OTS_NotLoaded If spells list is not loaded.
- **Since** 0.0.1
- **Since** 0.0.7
- **Access** public

PDO function POT::getDBHandle() [*line 560*]

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

OTS_HousesList function POT::getHousesList() [*line 1114*]

Returns list of laoded houses.

Returns list of laoded houses.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If houses list is not loaded.
- **Since** 0.1.0+SVN
- **Since** 0.0.1
- **Access** public

POT function POT::getInstance() [*line 316*]

Singleton.

Singleton.

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

OTS_Spell|null function POT::getInstant(\$name) [*line 1007*]

Function Parameters:

- *string* **\$name** Spell name.

Returns given instant spell.

Returns given instant spell.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Deprecated** 0.1.0+SVN Use POT::getSpellsList()->getInstant().
- **Throws** E_OTS_NotLoaded If spells list is not loaded.
- **Since** 0.0.1
- **Since** 0.0.7
- **Access** public

array function POT::getInstantsList() [*line 985*]

Returns list of instants.

Returns list of instants.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Deprecated** 0.1.0+SVN Use POT::getSpellsList()->getInstantsList().
- **Throws** E_OTS_NotLoaded If spells list is not loaded.
- **Since** 0.0.1
- **Since** 0.0.7
- **Access** public

OTS_ItemsList function POT::getItemsList() [*line 1205*]

Returns list of laoded items.

Returns list of laoded items.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If items list is not loaded.
- **Since** 0.1.0+SVN
- **Since** 0.0.1
- **Access** public

OTS_OTBMFile function POT::getMap() [*line 1299*]

Returns loaded map.

Returns loaded map.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If map is not loaded.
- **Since** 0.1.0+SVN
- **Since** 0.0.1
- **Access** public

OTS_Monster *null* function POT::getMonster(\$name) [*line 837*]
Function Parameters:

- *string* **\$name** Monster name.

Returns loaded data of given monster.
Returns loaded data of given monster.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Deprecated** 0.1.0+SVN Use POT::getMonstersList()->getMonster().
- **Throws** E_OTS_NotLoaded If monsters list is not loaded.
- **Since** 0.0.1
- **Since** 0.0.6
- **Access** public

OTS_MonstersList function POT::getMonstersList() [*line 857*]
Returns list of laoded monsters.
Returns list of laoded monsters.

- **Version** 0.1.0+SVN
- **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 964]

Function Parameters:

- *string* **\$name** Rune name.

Returns given rune spell.

Returns given rune spell.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Deprecated** 0.1.0+SVN Use POT::getSpellsList()->getRune().
- **Throws** E_OTS_NotLoaded If spells list is not loaded.
- **Since** 0.0.1
- **Since** 0.0.7
- **Access** public

array function POT::getRunesList() [line 942]

Returns list of runes.

Returns list of runes.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Deprecated** 0.1.0+SVN Use POT::getSpellsList()->getRunesList().
- **Throws** E_OTS_NotLoaded If spells list is not loaded.
- **Since** 0.0.1
- **Since** 0.0.7
- **Access** public

OTS_SpellsList function POT::getSpellsList() [*line 921*]

Returns list of laoded spells.

Returns list of laoded spells.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If spells list is not loaded.
- **Since** 0.1.0+SVN
- **Since** 0.0.1
- **Access** public

int|bool function POT::getVocationId(\$name) [*line 729*]

Function Parameters:

- *string* **\$name** Vocation.

Returns vocation's ID.

Returns vocation's ID.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Deprecated** 0.1.0+SVN Use POT::getVocationsList()->getVocationId().
- **Throws** E_OTS_NotLoaded If vocations list is not loaded.
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

string|bool function POT::getVocationName(\$id) [*line 751*]

Function Parameters:

- *int* **\$id** Vocation ID.

Returns name of given vocation's ID.

Returns name of given vocation's ID.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Deprecated** 0.1.0+SVN Use POT::getVocationsList()->getVocationName().
- **Throws** E_OTS_NotLoaded If vocations list is not loaded.
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

OTS_VocationsList function POT::getVocationsList() [*line 771*]

Returns vocations list object.

Returns vocations list object.

- **Version** 0.1.0+SVN
- **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::isIPBanned(\$ip) [*line 644*]

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 1275*]

Checks if OTBM is loaded.

Checks if OTBM is loaded.

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

void function POT::loadClass(\$class) [line 391]
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 1078]
Function Parameters:

- *string* **\$path** Houses file.

Loads houses list file.

Loads houses list file.

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

void function POT::loadItems(\$path) [line 1161]

Function Parameters:

- *string* **\$path** Items information directory.

Loads items list.

Loads items list.

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

void function POT::loadMap(\$path) [line 1254]

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

void function POT::loadMonsters(\$path) [line 799]

Function Parameters:

- *string* **\$path** Monsters directory.

Loads monsters mapping file.

Loads monsters mapping file.

- **Version** 0.1.0+SVN
- **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+SVN
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.7
- **Access** public

void function POT::loadVocations(\$file) [*line 690*]

Function Parameters:

- *string* **\$file** vocations.xml file location.

Loads vocations list.

Loads vocations list.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

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::setItemsCache([$cache = null]) [line 1140]`

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 1231]`

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 347]`

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 611]`

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::unloadHouses() [line 1101]

Unloads houses list.

Unloads houses list.

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

void function POT::unloadItems() [line 1192]

Unloads items list.

Unloads items list.

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

void function POT::unloadMap() [*line 1286*]

Unloads OTBM map.

Unloads OTBM map.

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

void function POT::unloadMonsters() [*line 822*]

Unloads monsters list.

Unloads monsters list.

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

void function POT::unloadSpells() [*line 908*]

Unloads spells list.

Unloads spells list.

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

void function POT::unloadVocations() [*line 714*]

Unloads vocations list.

Unloads vocations list.

- **Version** 0.1.0+SVN
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0+SVN
- **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** 0.1.0+SVN
- **Version** 0.0.2
- **Copyright** 2007 (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_OTBMEError](#)

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_DAO

- [IOTS_DAO](#)

IOTS_DB

- [IOTS_DB](#)

IOTS_FileCache

- [IOTS_FileCache](#)
 - [IOTS_ItemsCache](#)

IOTS_GuildAction

- [IOTS_GuildAction](#)

OTS_Base_DAO

- [OTS_Base_DAO](#)
 - [OTS_Account](#)

- [OTS_Group](#)
- [OTS_Guild](#)
- [OTS_GuildRank](#)
- [OTS_House](#)
- [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_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_FileNode

- [OTS_FileNode](#)

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_Spell

- [OTS_Spell](#)

OTS_SpellsList

- [OTS_SpellsList](#)

OTS_SQLField

- [OTS_SQLField](#)

OTS_VocationsList

- [OTS_VocationsList](#)

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

[0.1.0+SVN]

- * 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>
- * Additional wrappers for resource binds. <wrzasq>
- * Some additional updates to match OTServ development. <wrzasq>
- * Fixed some typos. <wrzasq>
- * Code vleanup. <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 instanced 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 automaticly 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 instead of double ones.

' are, in standard way, faster than " and it keeps code cleaner if you simply concat everything rather than inserting something like placeholders into string.

[8] Use spaces between parenthesis 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 a lot of resources (relatively).

II. File naming:

[1] Use lowercase names for directories.

[2] Use files 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 0.1.0+SVN version?

- * Houses support.

It was mentioned to be in 0.0.9 release but as houses use both data/ directory resources and SQL database we wanted to avoid making big mess with reconstructing it later.

Houses support including XML data loading and SQL rent controll.

Also OTBM support has been extended to load house tiles and houses list from external XML file.

- * No need to call POT::createObject().

All classes which use database handle fetch it now by themselves, you just call standard class constructors.

Of course POT::createObject() method will stay to provide backward compatibility, but you don't need to call it anymore!

Same reffers to POT::createFilter() method.

- * Separated data/ directory resources loaders.

From now you can load each data/ directory in separated object. This allows you to load, for example, data from two different data/ directories pararelly.

- * Default data/ directory resources instances within POT class.

Also now in POT class each resource (event that which didn't have loading method until now) has default instance which indicates that this instance stores info about current server gameworld. It allows you to link information references stored in other objects as IDs with full data from files.

- * Each data/ directory resource reader class implements an Iterator/IteratorAggregate, Countable and ArrayAccess interfaces.

They provide interface to use them as an arrays both for iteration and for counting.

Also you can use their's fields like array fields.

- * Uses IteratorAggregate interface instead of Iterator in some places (returns ArrayIterator as iterator).

This makes our code smaller. However note that this cause direct call to `current()`, `rewind()` and other `Iterator` interface methods impossible on objects with `IteratorAggregate` interface implemented (for backward compatibility those methods are still there, but are marked as deprecated).

- * Additional wrappers for resource binds.

There are some additional functions which can make it easier for you to evaluate some things like create conjure item with `sepll` object etc.

- * Some additional updates to match `OTServ` development.

Like for example `ITEM_TYPE_TELEPORT` added etc.

Appendix D - Todo List

In Package POT

In [OTS.php](#)

- 0.1.0: Implement `__toString()` as display drivers.
- 0.1.1: Support for call constructors with ID/name parameter for automatic pre-load for data.
- 0.1.2: OTAdmin protocol.
- 0.1.3: SOAP interface for remote controll.
- 0.2.0: Implement `NetworkMessage`.
- 0.2.0: Implement `OutOfBoundsException` instead of mixed results types.
- 1.0.0: Complete `phpUnit` test.
- 1.0.0: Create automatic wrappers for related OTS resource handlers.
- 1.0.0: Deprecations cleanup.
- 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).

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_Spell::__construct()	346
<i>Sets spell info.</i>	
constructor OTS_MonstersList::__construct()	270
<i>Loads monsters mapping file.</i>	
constructor OTS_MapCoords::__construct()	259
<i>Sets coords for point.</i>	
constructor OTS_ItemType::__construct()	247
<i>Initializes new item type object.</i>	
constructor OTS_SpellsList::__construct()	356
<i>Loads spells list.</i>	
constructor OTS_SQLField::__construct()	361
<i>Creates new field representation.</i>	
CHANGELOG	426
compat.php	415
<i>POT compatibility assurance package.</i>	
constructor OTS_VocationsList::__construct()	370
<i>Loads vocations list.</i>	
constructor OTS_Item::__construct()	214
<i>Creates item of given ID.</i>	
constructor OTS_HousesList::__construct()	200
<i>Loads houses information.</i>	
constructor OTS_Base_List::__construct()	112
<i>Sets database connection handler.</i>	
constructor OTS_Base_DAO::__construct()	108
<i>Sets database connection handler.</i>	
constructor IOTS_GuildAction::__construct()	87
<i>Objects are initialized with a guild that they are assigned to.</i>	
constructor IOTS_DB::__construct()	82
<i>Connection parameters.</i>	
constructor OTS_DB_MySQL::__construct()	123
<i>Creates database connection.</i>	
constructor OTS_DB_ODBC::__construct()	127
<i>Creates database connection.</i>	
constructor OTS_House::__construct()	192
<i>Creates wrapper for given house element.</i>	
constructor OTS_DB_SQLite::__construct()	134
<i>Creates database connection.</i>	
constructor OTS_DB_PostgreSQL::__construct()	130

Creates database connection.	
constructor E_OTS_ErrorCode::construct()	74
Sets error code.	

D

DAO objects	9
-----------------------------	---

E

E_OTS_NoDriver	78
Occurs when code attempts to execute driven action that has no assigned driver to handle it.	
E_OTS_NotAContainer	78
Occurs when in database item which is not a container contains sub-items.	
E_OTS_FileLoaderError::ERROR_TELL_ERROR	77
Failed to read position in file.	
E_OTS_FileLoaderError::ERROR_SEEK_ERROR	77
Failed to seek in given position in file.	
E_OTS_FileLoaderError::ERROR_NOT_OPEN	76
Attempted to execute operation on not opened file.	
E_OTS_NotLoaded	79
Occurs when code attempts to access property of not loaded object.	
E_OTS_OTBMErrors	79
OTBM map loading error.	
E_OTS_ReadOnly	81
Occurs when code attempts to evaluate write operation on read-only object.	
E_OTS_OutOfBuffer	80
Occurs when properties stream has ended and there is still read attempt.	
E_OTS_OTBMErrors::LOADMAPERROR_UNKNOWNNODETYPE	80
Unknown node type.	
E_OTS_OTBMErrors::LOADMAPERROR_OUTDATEDHEADER	79
Unsupported file version.	
E_OTS_FileLoaderError::ERROR_INVALID_FORMAT	76
File corrupted.	
E_OTS_FileLoaderError::ERROR_INVALID_FILE_VERSION	76
Unsupported file version.	
E_OTS_NotLoaded.php	29
E_OTS_OTBMErrors.php	30
Code in this file bases on oryiginal OTServ OTBM format loading C++ code (iomapotbm).	
E_OTS_NotAContainer.php	28
E_OTS_NoDriver.php	27
E_OTS_FileLoaderError.php	26
Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader).	
E_OTS_OutOfBuffer.php	31
Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader).	
E_OTS_ReadOnly.php	32
E_OTS_FileLoaderError::ERROR_EOF	75
Unexpected end of file.	
E_OTS_FileLoaderError::ERROR_CAN_NOT_OPEN	75
Could not open file.	
E_OTS_FileLoaderError	75

<i>Error during reading OTServ binary file.</i>	
E_OTSErrorCode	74
<i>Generic exception class for error codes.</i>	
E_OTSErrorCode.php	25

G

Guild action drivers	15
Guilds	13

I

IOTS_GuildAction::addRequest()	88
<i>Adds new request.</i>	
IOTS_GuildAction::deleteRequest()	88
<i>Deletes request.</i>	
IOTS_GuildAction	87
<i>Guild action interface.</i>	
IOTS_FileCache::writeCache()	86
<i>Writes node cache.</i>	
IOTS_FileCache::readCache()	85
<i>Returns cache.</i>	
IOTS_GuildAction::listRequests()	88
<i>List of saved pending actions.</i>	
IOTS_GuildAction::submitRequest()	89
<i>Finalizes request.</i>	
INSTALL	431
IOTS_ItemsCache::writeItems()	90
<i>Writes items cache.</i>	
IOTS_ItemsCache::readItems()	90
<i>Returns cache.</i>	
IOTS_ItemsCache	89
<i>This interface defines items.</i>	
IOTS_FileCache	85
<i>This interface describes binary files cache control drivers.</i>	
IOTS_DB::tableName()	85
<i>Query-quoted table name.</i>	
IOTS_ItemsCache.php	37
IOTS_DAO	81
<i>OTServ database object.</i>	
IOTS_GuildAction.php	36
IOTS_FileCache.php	35
IOTS_DB.php	34
IOTS_DB	81
<i>OTServ database handler interface.</i>	
IOTS_DB::fieldName()	82
<i>Query-quoted field name.</i>	
IOTS_DB::SQLquote()	84
<i>Query-quoted string value.</i>	
IOTS_DB::SQLquery()	84
<i>Evaluates query.</i>	

IOTS_DB::limit()	83
<i>LIMIT/OFFSET clause for queries.</i>	
IOTS_DB::lastInsertId()	83
<i>ID of last created record.</i>	
IOTS.DAO.php	33

L

LICENSE	424
-------------------------	-----

N

NEWS	431
----------------------	-----

O

OTS_MonstersList::set_state()	275
<i>Magic PHP5 method.</i>	
OTS_OTBMFile	275
<i>OTBM format reader.</i>	
OTS_OTBMFile::OTBM_ATTR_ACTION_ID	276
<i>Action ID.</i>	
OTS_OTBMFile::OTBM_ATTR_DEPOT_ID	276
<i>Depot ID.</i>	
OTS_MonstersList::valid()	274
<i>Checks if there is anything more in iterator.</i>	
OTS_MonstersList::rewind()	274
<i>Resets iterator index.</i>	
OTS_MonstersList::offsetGet()	272
<i>Returns item from given position.</i>	
OTS_MonstersList::offsetSet()	273
<i>This method is implemented for ArrayAccess interface.</i>	
OTS_MonstersList::offsetUnset()	273
<i>This method is implemented for ArrayAccess interface.</i>	
OTS_OTBMFile::OTBM_ATTR_DESC	276
<i>Description.</i>	
OTS_OTBMFile::OTBM_ATTR_DESCRIPTION	277
<i>Description attribute.</i>	
OTS_OTBMFile::OTBM_ATTR_RUNE_CHARGES	279
<i>Rune charges amount.</i>	
OTS_OTBMFile::OTBM_ATTR_TELE_DEST	279
<i>Teleport destination.</i>	
OTS_OTBMFile::OTBM_ATTR_TEXT	280
<i>Text.</i>	
OTS_OTBMFile::OTBM_ATTR_TILE_FLAGS	280
<i>Tile flags.</i>	
OTS_OTBMFile::OTBM_ATTR_ITEM	279
<i>Item.</i>	
OTS_OTBMFile::OTBM_ATTR_HOUSEDOORID	278
<i>ID of doors.</i>	

OTS OTBMFile::OTBM_ATTR_EXT_FILE	277
<i>External file.</i>	
OTS OTBMFile::OTBM_ATTR_EXT_HOUSE_FILE	277
<i>External houses file.</i>	
OTS OTBMFile::OTBM_ATTR_EXT_SPAWN_FILE	278
<i>External spawns file.</i>	
OTS MonstersList::offsetExists()	272
<i>Checks if given element exists.</i>	
OTS MonstersList::next()	272
<i>Moves to next iterator monster.</i>	
OTS Monster::getItems()	266
<i>Returns all possible loot.</i>	
OTS Monster::getLoot()	266
<i>Returns all possible loot.</i>	
OTS Monster::getManaCost()	267
<i>Returns amount of mana required to summon this monster.</i>	
OTS Monster::getName()	267
<i>Returns monster name.</i>	
OTS Monster::getImmunities()	266
<i>Returns all monster immunities.</i>	
OTS Monster::getHealth()	265
<i>Returns monster HP.</i>	
OTS Monster::getExperience()	264
<i>Returns amount of experience for killing this monster.</i>	
OTS Monster::getFlag()	265
<i>Returns specified flag value.</i>	
OTS Monster::getFlags()	265
<i>Returns all monster flags (in format flagname => value).</i>	
OTS Monster::getRace()	267
<i>Returns monster race.</i>	
OTS Monster::getSpeed()	268
<i>Returns monster speed.</i>	
OTS MonstersList::current()	271
<i>Returns monster at current position in iterator.</i>	
OTS MonstersList::getMonster()	271
<i>Returns loaded data of given monster.</i>	
OTS MonstersList::key()	271
<i>Returns name of current position.</i>	
OTS MonstersList::count()	270
<i>Returns amount of monsters loaded.</i>	
OTS MonstersList	270
<i>Wrapper for monsters list.</i>	
OTS Monster::getVoices()	268
<i>Returns voices that monster can sound.</i>	
OTS Monster::hasImmunity()	268
<i>Checks if monster has given immunity.</i>	
OTS Monster::get()	269
<i>Magic PHP5 method.</i>	
OTS OTBMFile::OTBM_ATTR_UNIQUE_ID	280
<i>Unique ID.</i>	
OTS OTBMFile::OTBM_NODE_HOUSEDILE	281
<i>Tile of house.</i>	
OTS OTBMFile::next()	290

<i>Moves to next iterator town.</i>	
OTS_OTBMFile::offsetExists()	291
<i>Checks if given element exists.</i>	
OTS_OTBMFile::offsetGet()	291
<i>Returns item from given position.</i>	
OTS_OTBMFile::offsetSet()	292
<i>This method is implemented for ArrayAccess interface.</i>	
OTS_OTBMFile::loadFile()	290
<i>Loads OTBM file content.</i>	
OTS_OTBMFile::key()	290
<i>Returns ID of current position.</i>	
OTS_OTBMFile::getTownsList()	288
<i>Returns list (id => name) of loaded towns.</i>	
OTS_OTBMFile::getTownTemple()	289
<i>Returns town's temple position.</i>	
OTS_OTBMFile::getWidth()	289
<i>Returns map width.</i>	
OTS_OTBMFile::offsetUnset()	293
<i>This method is implemented for ArrayAccess interface.</i>	
OTS_OTBMFile::rewind()	293
<i>Resets iterator index.</i>	
OTS_Player::ban()	297
<i>Bans current player.</i>	
OTS_Player::delete()	298
<i>Deletes player.</i>	
OTS_Player::find()	298
<i>Loads player by it's name.</i>	
OTS_Player::getAccount()	299
<i>Returns account of this player.</i>	
OTS_Player	295
<i>OTServ character abstraction.</i>	
OTS_OTBMFile::__wakeup()	295
<i>Magic PHP5 method.</i>	
OTS_OTBMFile::valid()	294
<i>Checks if there is anything more in iterator.</i>	
OTS_OTBMFile::__get()	294
<i>Magic PHP5 method.</i>	
OTS_OTBMFile::__set_state()	295
<i>Magic PHP5 method.</i>	
OTS_OTBMFile::getTownName()	288
<i>Returns name of given town's ID.</i>	
OTS_OTBMFile::getTownID()	287
<i>Returns town's ID.</i>	
OTS_OTBMFile::OTBM_NODE_SPAWNS	283
<i>Spawns container.</i>	
OTS_OTBMFile::OTBM_NODE_SPAWN_AREA	283
<i>Spawn.</i>	
OTS_OTBMFile::OTBM_NODE_TILE	283
<i>Single tile.</i>	
OTS_OTBMFile::OTBM_NODE_TILE_AREA	284
<i>Map tiles fragment.</i>	
OTS_OTBMFile::OTBM_NODE_ROOTV1	282
<i>Root node.</i>	

OTS OTBMFile::OTBM_NODE MONSTER	282
<i>Monster.</i>	
OTS OTBMFile::OTBM_NODE ITEM	281
<i>Item.</i>	
OTS OTBMFile::OTBM_NODE ITEM_DEF	281
<i>Item definition.</i>	
OTS OTBMFile::OTBM_NODE MAP_DATA	282
<i>Map data container.</i>	
OTS OTBMFile::OTBM_NODE TILE_REF	284
<i>Tile reference.</i>	
OTS OTBMFile::OTBM_NODE TILE_SQUARE	284
<i>Tile.</i>	
OTS OTBMFile::getHeight()	286
<i>Returns map height.</i>	
OTS OTBMFile::getHousesList()	287
<i>Loads map's houses list.</i>	
OTS OTBMFile::getIterator()	287
<i>Returns iterator handle for loops.</i>	
OTS OTBMFile::getDescription()	286
<i>Returns map description.</i>	
OTS OTBMFile::current()	286
<i>Returns town at current position in iterator.</i>	
OTS OTBMFile::OTBM_NODE TOWN	285
<i>Town.</i>	
OTS OTBMFile::OTBM_NODE TOWNS	285
<i>Towns container.</i>	
OTS OTBMFile::count()	285
<i>Returns amount of towns loaded.</i>	
OTS Monster::getDefenses()	264
<i>Returns list of special defenses.</i>	
OTS Monster::getDefense()	264
<i>Returns monster defense rate.</i>	
OTS ItemType::ITEM_GROUP CONTAINER	240
<i>Container.</i>	
OTS ItemType::ITEM_GROUP DOOR	241
<i>Door.</i>	
OTS ItemType::ITEM_GROUP FLUID	241
<i>Liquid thing.</i>	
OTS ItemType::ITEM_GROUP GROUND	241
<i>Ground tile.</i>	
OTS ItemType::ITEM_GROUP ARMOR	240
<i>Armor.</i>	
OTS ItemType::ITEM_GROUP AMMUNITION	240
<i>Ammunition.</i>	
OTS ItemType::FLAG STACKABLE	239
<i>Can be grouped with another items.</i>	
OTS ItemType::FLAG USEABLE	239
<i>Can be used by players.</i>	
OTS ItemType::FLAG VERTICAL	239
<i>Is oriented vertically.</i>	
OTS ItemType::ITEM_GROUP KEY	242
<i>Key.</i>	
OTS ItemType::ITEM_GROUP MAGICFIELD	242

<i>Magic field.</i>	
OTS_ItemType::ITEM_GROUP_WRITEABLE	244
<i>Item that can store editable sign.</i>	
OTS_ItemType::ITEM_TYPE_CONTAINER	245
<i>Container.</i>	
OTS_ItemType::ITEM_TYPE_DEPOT	245
<i>Depot locker.</i>	
OTS_ItemType::ITEM_TYPE_DOOR	245
<i>Door.</i>	
OTS_ItemType::ITEM_GROUP_WEAPON	244
<i>Weapon.</i>	
OTS_ItemType::ITEM_GROUP_TELEPORT	244
<i>Teleport field.</i>	
OTS_ItemType::ITEM_GROUP_NONE	242
<i>No group speciffied.</i>	
OTS_ItemType::ITEM_GROUP_RUNE	243
<i>Rune.</i>	
OTS_ItemType::ITEM_GROUP_SPLASH	243
<i>Splash effect.</i>	
OTS_ItemType::FLAG_ROTABLE	238
<i>Can be rotated by player.</i>	
OTS_ItemType::FLAG_READABLE	238
<i>Has readable sign.</i>	
OTS_ItemType::FLAG_BLOCK_PATHFIND	233
<i>Can block searching for path.</i>	
OTS_ItemType::FLAG_BLOCK_PROJECTILE	234
<i>BLOCK_PROJECTILE flag(?).</i>	
OTS_ItemType::FLAG_BLOCK_SOLID	234
<i>Can block characters from walking.</i>	
OTS_ItemType::FLAG_CANNOTDECAY	234
<i>Doesn't decay.</i>	
OTS_ItemType::FLAG_ALWAYSONTOP	233
<i>Is always over other items in stack.</i>	
OTS_ItemType::FLAG_ALLOWDISTREAD	233
<i>Can be read from distance.</i>	
OTS_ItemsList:: set_state()	231
<i>Magic PHP5 method.</i>	
OTS_ItemsList:: wakeup()	231
<i>Magic PHP5 method.</i>	
OTS_ItemType	232
<i>Item type info.</i>	
OTS_ItemType::FLAG_FLOORCHANGEDOWN	235
<i>Changes floor under it.</i>	
OTS_ItemType::FLAG_FLOORCHANGEEST	235
<i>Changes floor east from it's position.</i>	
OTS_ItemType::FLAG_HORIZONTAL	237
<i>Is oriented horizontally.</i>	
OTS_ItemType::FLAG_MOVEABLE	237
<i>Can be moved by player.</i>	
OTS_ItemType::FLAG_PICKUPABLE	238
<i>Can be picked up by player.</i>	
OTS_ItemType::FLAG_HAS_HEIGHT	237
<i>Does item rises stack height on it's field.</i>	

OTS ItemType::FLAG_HANGABLE	236
<i>Can be hang(?).</i>	
OTS ItemType::FLAG_FLOORCHANGENORTH	235
<i>Changes floor north from it's position.</i>	
OTS ItemType::FLAG_FLOORCHANGESOUTH	236
<i>Changes floor south from it's position.</i>	
OTS ItemType::FLAG_FLOORCHANGEWEST	236
<i>Changes floor west from it's position.</i>	
OTS ItemType::ITEM_TYPE_MAGICFIELD	246
<i>Magic field.</i>	
OTS ItemType::ITEM_TYPE_MAILBOX	246
<i>Mailbox.</i>	
OTS ItemType::setGroup()	256
<i>Sets item group.</i>	
OTS ItemType::setName()	256
<i>Sets item type name.</i>	
OTS ItemType::setType()	257
<i>Sets item type.</i>	
OTS ItemType::__get()	257
<i>Magic PHP5 method.</i>	
OTS ItemType::setFlags()	255
<i>Sets type flags.</i>	
OTS ItemType::setClientId()	255
<i>Sets client side ID.</i>	
OTS ItemType::isUsable()	254
<i>Checks if item is usable.</i>	
OTS ItemType::isVertical()	254
<i>Checks if item is vertical.</i>	
OTS ItemType::setAttribute()	254
<i>Sets given attribute.</i>	
OTS ItemType::__set()	258
<i>Magic PHP5 method.</i>	
OTS ItemType::__set_state()	258
<i>Magic PHP5 method.</i>	
OTS MapCoords::__set_state()	261
<i>Magic PHP5 method.</i>	
OTS Monster	262
<i>Wrapper for monsters files DOMDocument.</i>	
OTS Monster::getArmor()	263
<i>Returns monster armor.</i>	
OTS Monster::getAttacks()	263
<i>Returns list of monster attacks.</i>	
OTS MapCoords::__get()	261
<i>Magic PHP5 method.</i>	
OTS MapCoords::getZ()	261
<i>Returns Z.</i>	
OTS MapCoords	259
<i>Map position point.</i>	
OTS MapCoords::getX()	260
<i>Returns X.</i>	
OTS MapCoords::getY()	260
<i>Returns Y.</i>	
OTS ItemType::isStackable()	253

<i>Checks if item is stackable.</i>	
OTS_ItemType::isRotable()	253
<i>Checks if item can be rotated.</i>	
OTS_ItemType::getAttributesList()	249
<i>Returns all attributes list.</i>	
OTS_ItemType::getClientId()	249
<i>Returns item type client ID.</i>	
OTS_ItemType::getGroup()	249
<i>Returns group.</i>	
OTS_ItemType::getId()	250
<i>Returns item type server ID.</i>	
OTS_ItemType::getAttribute()	248
<i>Returns given attribute.</i>	
OTS_ItemType::createItem()	248
<i>Creates instance of this type.</i>	
OTS_ItemType::ITEM_TYPE_NONE	246
<i>No special type.</i>	
OTS_ItemType::ITEM_TYPE_TELEPORT	247
<i>Teleport.</i>	
OTS_ItemType::ITEM_TYPE_TRASHHOLDER	247
<i>Trash can.</i>	
OTS_ItemType::getName()	250
<i>Returns item name.</i>	
OTS_ItemType::getType()	250
<i>Returns item type.</i>	
OTS_ItemType::isMovable()	252
<i>Checks if item is movable.</i>	
OTS_ItemType::isPickupable()	252
<i>Checks if item is pickupable.</i>	
OTS_ItemType::isReadable()	253
<i>Checks if item is readable.</i>	
OTS_ItemType::isHorizontal()	252
<i>Checks if item is horizontal.</i>	
OTS_ItemType::isHangable()	251
<i>Checks if item can be hanged.</i>	
OTS_ItemType::hasHeight()	250
<i>Checks if item has height.</i>	
OTS_ItemType::isAlwaysOnTop()	251
<i>Checks if item is always on top.</i>	
OTS_ItemType::isBlocking()	251
<i>Checks if item is blocking.</i>	
OTS_Player::getCap()	299
<i>Capacity.</i>	
OTS_Player::getConditions()	299
<i>Conditions.</i>	
OTS_Spell::getName()	350
<i>Returns spell name.</i>	
OTS_Spell::getReagent()	350
<i>Returns item type of reagent item.</i>	
OTS_Spell::getReagentId()	350
<i>Returns ID of item that is used by spell.</i>	
OTS_Spell::getSoul()	351
<i>Soul points cost.</i>	

OTS_Spell::getMana()	349
<i>Mana cost.</i>	
OTS_Spell::getMagicLevel()	349
<i>Magic level required to cast.</i>	
OTS_Spell::getConjureId()	348
<i>Returns ID of item conjured by this spell.</i>	
OTS_Spell::getId()	348
<i>Returns rune item id.</i>	
OTS_Spell::getLevel()	349
<i>Level required for use.</i>	
OTS_Spell::getType()	351
<i>Returns spell type.</i>	
OTS_Spell::getVocations()	351
<i>Returns list of vocations that are allowed to learn this spell.</i>	
OTS_Spell::isFarUseAllowed()	353
<i>Checks if distance use allowed.</i>	
OTS_Spell::isLearnNeeded()	354
<i>Checks if spell needs to be learned.</i>	
OTS_Spell::isPremium()	354
<i>Checks if spell requires PACC.</i>	
OTS_Spell::__get()	354
<i>Magic PHP5 method.</i>	
OTS_Spell::isEnabled()	353
<i>Checks if spell is enabled.</i>	
OTS_Spell::isAggressive()	353
<i>Checks if spell is threatened as unfriendly by other creatures.</i>	
OTS_Spell::getWords()	352
<i>Returns spell formula.</i>	
OTS_Spell::hasParams()	352
<i>Checks if spell has parameter.</i>	
OTS_Spell::isAggressive()	352
<i>This method is the same as OTS_Spell::isAggressive().</i>	
OTS_Spell::getConjureCount()	348
<i>Returns amount of items conjured by this spell.</i>	
OTS_Spell::getConjure()	347
<i>Returns item type of conjured item.</i>	
OTS_Player::setTownId()	340
<i>Sets residence town's ID.</i>	
OTS_Player::setVocation()	340
<i>Sets player profession.</i>	
OTS_Player::unban()	341
<i>Deletes ban from current player.</i>	
OTS_Player::unsetRedSkull()	341
<i>Unsets red skull flag.</i>	
OTS_Player::setSoul()	340
<i>Sets soul points.</i>	
OTS_Player::setSlot()	339
<i>Sets slot content.</i>	
OTS_Player::setSex()	337
<i>Sets player gender.</i>	
OTS_Player::setSkill()	338
<i>Sets skill value.</i>	
OTS_Player::setSkillTries()	338

<i>Sets skill's tries for next level.</i>	
OTS_Player::unsetSave()	342
<i>Unsets save flag.</i>	
OTS_Player::__get()	342
<i>Magic PHP5 method.</i>	
OTS_Spell	345
<i>Wrapper for spell info.</i>	
OTS_Spell::createConjure()	346
<i>Creates conjure item.</i>	
OTS_Spell::getCharges()	347
<i>Number of rune charges.</i>	
OTS_Players_List::init()	344
<i>Sets list parameters.</i>	
OTS_Players_List::deletePlayer()	344
<i>Deletes player.</i>	
OTS_Player::__set()	343
<i>Magic PHP5 method.</i>	
OTS_Player::__sleep()	343
<i>Magic PHP5 method.</i>	
OTS_Players_List	344
<i>List of players.</i>	
OTS_SpellsList	355
<i>Wrapper for spells list.</i>	
OTS_SpellsList::SPELL_CONJURE	355
<i>Conjure spell.</i>	
OTS_SQLFilter::compareField()	367
<i>Compares field with a literal value.</i>	
OTS_SQLFilter::getTables()	368
<i>Returns list of all tables used by filter.</i>	
OTS_SQLFilter::__get()	368
<i>Magic PHP5 method.</i>	
OTS_SQLFilter::__sleep()	368
<i>Magic PHP5 method.</i>	
OTS_SQLFilter::addFilter()	366
<i>General-purpose filter.</i>	
OTS_SQLFilter::OPERATOR_NLOWER	366
<i>Not-lower-then operator.</i>	
OTS_SQLFilter::OPERATOR_NEQUAL	365
<i>Not-equal operator.</i>	
OTS_SQLFilter::OPERATOR_NGREATER	365
<i>Not-greater-then operator.</i>	
OTS_SQLFilter::OPERATOR_NLIKE	366
<i>Not-LIKE operator.</i>	
OTS_SQLFilter::__toString()	369
<i>Returns string representation of WHERE clause.</i>	
OTS_VocationsList	369
<i>Wrapper for vocations.</i>	
OTS_VocationsList::offsetGet()	372
<i>Returns item from given position.</i>	
OTS_VocationsList::offsetSet()	373
<i>This method is implemented for ArrayAccess interface.</i>	
OTS_VocationsList::offsetUnset()	373
<i>This method is implemented for ArrayAccess interface.</i>	

OTS_VocationsList:: set_state()	374
<i>Magic PHP5 method.</i>	
OTS_VocationsList::offsetExists()	372
<i>Checks if given element exists.</i>	
OTS_VocationsList::getVocationName()	371
<i>Returns name of given vocation's ID.</i>	
OTS_VocationsList::count()	370
<i>Returns amount of vocations loaded.</i>	
OTS_VocationsList::getIterator()	370
<i>Returns iterator handle for loops.</i>	
OTS_VocationsList::getVocationId()	371
<i>Returns vocation's ID.</i>	
OTS_SQLFilter::OPERATOR_LOWER	365
<i>Lower-than operator.</i>	
OTS_SQLFilter::OPERATOR LIKE	364
<i>LIKE operator.</i>	
OTS_SpellsList::getInstantsList()	358
<i>Returns list of instants.</i>	
OTS_SpellsList::getRune()	358
<i>Returns given rune spell.</i>	
OTS_SpellsList::getRunesList()	359
<i>Returns list of runes.</i>	
OTS_SpellsList:: get()	359
<i>Magic PHP5 method.</i>	
OTS_SpellsList::getInstant()	358
<i>Returns given instant spell.</i>	
OTS_SpellsList::getConjuresList()	357
<i>Returns list of conjure spells.</i>	
OTS_SpellsList::SPELL_INSTANT	356
<i>Instant spell.</i>	
OTS_SpellsList::SPELL_RUNE	356
<i>Rune spell.</i>	
OTS_SpellsList::getConjure()	357
<i>Returns given conjure spell.</i>	
OTS_SpellsList:: set_state()	360
<i>Magic PHP5 method.</i>	
OTS_SQLField	360
<i>SQL identifier representation.</i>	
OTS_SQLFilter::CRITERIUM_OR	363
<i>OR sibling.</i>	
OTS_SQLFilter::OPERATOR_EQUAL	364
<i>Equal operator.</i>	
OTS_SQLFilter::OPERATOR GREATER	364
<i>Greater-than operator.</i>	
OTS_SQLFilter::CRITERIUM AND	363
<i>AND sibling.</i>	
OTS_SQLFilter	363
<i>SQL WHERE clause object.</i>	
OTS_SQLField::getName()	361
<i>Returns field name.</i>	
OTS_SQLField::getTable()	362
<i>Returns table name.</i>	
OTS_SQLField:: get()	362

<i>Magic PHP5 method.</i>	
OTS_Player::setSave()	337
<i>Sets save flag.</i>	
OTS_Player::setRedSkullTime()	336
<i>Sets red skulled time remained.</i>	
OTS_Player::getPosX()	311
<i>X map coordinate.</i>	
OTS_Player::getPosY()	311
<i>Y map coordinate.</i>	
OTS_Player::getPosZ()	311
<i>Z map coordinate.</i>	
OTS_Player::getPremiumEnd()	312
<i>Player's Premium Account expiration timestamp.</i>	
OTS_Player::getName()	310
<i>Player name.</i>	
OTS_Player::getManaSpent()	310
<i>Mana spent.</i>	
OTS_Player::getMagLevel()	309
<i>Magic level.</i>	
OTS_Player::getMana()	309
<i>Current mana.</i>	
OTS_Player::getManaMax()	309
<i>Maximum mana.</i>	
OTS_Player::getRank()	312
<i>Assigned guild rank.</i>	
OTS_Player::getRankId()	313
<i>Guild rank ID.</i>	
OTS_Player::getSlot()	315
<i>Returns items tree from given slot.</i>	
OTS_Player::getSoul()	316
<i>Soul points.</i>	
OTS_Player::getTownId()	316
<i>Residence town's ID.</i>	
OTS_Player::getVocation()	317
<i>Player proffesion.</i>	
OTS_Player::getSkillTries()	315
<i>Returns player's skill's tries for next level.</i>	
OTS_Player::getSkill()	314
<i>Returns player's skill.</i>	
OTS_Player::getRedSkullTime()	313
<i>Red skulled time remained.</i>	
OTS_Player::getSave()	313
<i>Save counter.</i>	
OTS_Player::getSex()	314
<i>Player gender.</i>	
OTS_Player::getLossSkills()	308
<i>Percentage of skills lost after dead.</i>	
OTS_Player::getLossMana()	308
<i>Percentage of used mana lost after dead.</i>	
OTS_Player::getGuildNick()	302
<i>Guild nick.</i>	
OTS_Player::getHealth()	303
<i>Current HP.</i>	

OTS_Player::getHealthMax()	303
<i>Maximum HP.</i>	
OTS_Player::getId()	304
<i>Player ID.</i>	
OTS_Player::getGroup()	302
<i>Returns group of this player.</i>	
OTS_Player::getExperience()	302
<i>Experience points.</i>	
OTS_Player::getCustomField()	300
<i>Reads custom field.</i>	
OTS_Player::getDepot()	301
<i>Returns items tree from given depot.</i>	
OTS_Player::getDirection()	301
<i>Looking direction.</i>	
OTS_Player::getLastIP()	304
<i>Last login IP.</i>	
OTS_Player::getLastLogin()	304
<i>Last login timestamp.</i>	
OTS_Player::getLookLegs()	307
<i>Legs color.</i>	
OTS_Player::getLookType()	307
<i>Outfit.</i>	
OTS_Player::getLossExperience()	307
<i>Percentage of experience lost after dead.</i>	
OTS_Player::getLookHead()	306
<i>Hair color.</i>	
OTS_Player::getLookFeet()	306
<i>Boots color.</i>	
OTS_Player::getLevel()	305
<i>Experience level.</i>	
OTS_Player::getLookAddons()	305
<i>Addons.</i>	
OTS_Player::getLookBody()	305
<i>Body color.</i>	
OTS_Player::getVocationName()	317
<i>Player profession name.</i>	
OTS_Player::hasRedSkull()	318
<i>Checks if player has red skull.</i>	
OTS_Player::setLossMana()	330
<i>Sets percentage of used mana lost after dead.</i>	
OTS_Player::setLossSkills()	331
<i>Sets percentage of skills lost after dead.</i>	
OTS_Player::setMagLevel()	331
<i>Sets magic level.</i>	
OTS_Player::setMana()	331
<i>Sets current mana.</i>	
OTS_Player::setLossExperience()	330
<i>Sets percentage of experience lost after dead.</i>	
OTS_Player::setLookType()	329
<i>Sets outfit.</i>	
OTS_Player::setLookFeet()	328
<i>Sets boots color.</i>	
OTS_Player::setLookHead()	328

<i>Sets hair color.</i>	
OTS_Player::setLookLegs()	329
<i>Sets legs color.</i>	
OTS_Player::setManaMax()	332
<i>Sets maximum mana.</i>	
OTS_Player::setManaSpent()	332
<i>Sets mana spent.</i>	
OTS_Player::setRank()	335
<i>Assigns guild rank.</i>	
OTS_Player::setRankId()	336
<i>Sets guild rank ID.</i>	
OTS_Player::setRedSkull()	336
<i>Sets red skull flag.</i>	
OTS_Player::setPremiumEnd()	335
<i>Sets player's Premium Account expiration timestamp.</i>	
OTS_Player::setPosZ()	334
<i>Sets Z map coordinate.</i>	
OTS_Player::setName()	333
<i>Sets players's name.</i>	
OTS_Player::setPosX()	333
<i>Sets X map coordinate.</i>	
OTS_Player::setPosY()	334
<i>Sets Y map coordinate.</i>	
OTS_Player::setLookBody()	327
<i>Sets body color.</i>	
OTS_Player::setLookAddons()	327
<i>Sets addons.</i>	
OTS_Player::setAccount()	320
<i>Assigns character to account.</i>	
OTS_Player::setCap()	320
<i>Sets capacity.</i>	
OTS_Player::setConditions()	321
<i>Sets conditions.</i>	
OTS_Player::setCustomField()	321
<i>Writes custom field.</i>	
OTS_Player::save()	320
<i>Saves player in database.</i>	
OTS_Player::load()	319
<i>Loads player with given id.</i>	
OTS_Player::isBanned()	318
<i>Checks if player is banned.</i>	
OTS_Player::isLoading()	318
<i>Checks if object is loaded.</i>	
OTS_Player::isSaveSet()	319
<i>Checks if save flag is set.</i>	
OTS_Player::setDepot()	322
<i>Sets depot content.</i>	
OTS_Player::setDirection()	323
<i>Sets looking direction.</i>	
OTS_Player::setLastIP()	326
<i>Sets last login IP.</i>	
OTS_Player::setLastLogin()	326
<i>Sets last login timestamp.</i>	

OTS_Player::setLevel()	326
<i>Sets experience level.</i>	
OTS_Player::setHealthMax()	325
<i>Sets maximum HP.</i>	
OTS_Player::setHealth()	325
<i>Sets current HP.</i>	
OTS_Player::setExperience()	323
<i>Sets experience points.</i>	
OTS_Player::setGroup()	324
<i>Assigns character to group.</i>	
OTS_Player::setGuildNick()	324
<i>Sets guild nick.</i>	
OTS_ItemsList::get()	230
<i>Magic PHP5 method.</i>	
OTS_ItemsList::valid()	230
<i>Checks if there is anything more in iterator.</i>	
OTS_DB_ODBC::limit()	128
<i>LIMIT/OFFSET clause for queries.</i>	
OTS_DB_ODBC::SQLquery()	128
<i>IOTS_DB method.</i>	
OTS_DB_ODBC::SQLquote()	129
<i>IOTS_DB method.</i>	
OTS_DB_ODBC::tableName()	129
<i>Query-quoted table name.</i>	
OTS_DB_ODBC::fieldName()	127
<i>Query-quoted field name.</i>	
OTS_DB_ODBC	126
<i>ODBC connection interface.</i>	
OTS_DB_MySQL::SQLquery()	125
<i>IOTS_DB method.</i>	
OTS_DB_MySQL::SQLquote()	125
<i>IOTS_DB method.</i>	
OTS_DB_MySQL::tableName()	126
<i>Query-quoted table name.</i>	
OTS_DB_PostgreSQL	130
<i>PostgreSQL connection interface.</i>	
OTS_DB_PostgreSQL::fieldName()	131
<i>Query-quoted field name.</i>	
OTS_DB_SQLite::fieldName()	135
<i>Query-quoted field name.</i>	
OTS_DB_SQLite::limit()	135
<i>LIMIT/OFFSET clause for queries.</i>	
OTS_DB_SQLite::SQLquery()	135
<i>IOTS_DB method.</i>	
OTS_DB_SQLite::SQLquote()	136
<i>IOTS_DB method.</i>	
OTS_DB_SQLite	134
<i>SQLite connection interface.</i>	
OTS_DB_PostgreSQL::tableName()	133
<i>Query-quoted table name.</i>	
OTS_DB_PostgreSQL::limit()	131
<i>LIMIT/OFFSET clause for queries.</i>	
OTS_DB_PostgreSQL::SQLquery()	132

IOTS_DB method.	
OTS_DB PostgreSQL::SQLquote()	132
IOTS_DB method.	
OTS_DB MySQL::limit()	124
LIMIT/OFFSET clause for queries.	
OTS_DB MySQL::fieldName()	124
Query-quoted field name.	
OTS_Base_List::__set_state()	117
Magic PHP5 method.	
OTS_Base_List::__sleep()	118
Magic PHP5 method.	
OTS_Base_List::__wakeup()	118
Magic PHP5 method.	
OTS_Container	119
Container item representation.	
OTS_Base_List::__set()	117
Magic PHP5 method.	
OTS_Base_List::valid()	116
Checks if there are any rows left.	
OTS_Base_List::setFilter()	115
Sets filter on list.	
OTS_Base_List::setLimit()	115
Sets LIMIT.	
OTS_Base_List::setOffset()	116
Sets OFFSET.	
OTS_Container::addItem()	119
Adds item to container.	
OTS_Container::count()	119
Number of items inside container.	
OTS_Container::rewind()	122
Resets internal items array pointer.	
OTS_Container::valid()	122
Checks if there are any items left.	
OTS_DB MySQL	123
MySQL connection interface.	
OTS_Container::removeItem()	121
Removes given item from current container.	
OTS_Container::next()	121
Moves to next item.	
OTS_Container::current()	120
Returns current item.	
OTS_Container::getIterator()	120
Returns iterator handle for loops.	
OTS_Container::key()	121
Current cursor position.	
OTS_DB SQLite::tableName()	137
Query-quoted table name.	
OTS_FileLoader	137
Universal OTServ binary formats reader.	
OTS_FileNode::__clone()	148
Creates clone of object.	
OTS_FileNode::__get()	149
Magic PHP5 method.	

OTS_FileNode:: set()	149
<i>Magic PHP5 method.</i>	
OTS_FileNode:: set_state()	150
<i>Magic PHP5 method.</i>	
OTS_FileNode::skip()	148
<i>Skips given amount of bytes.</i>	
OTS_FileNode::setType()	147
<i>Sets node type.</i>	
OTS_FileNode::setBuffer()	146
<i>Sets properties stream.</i>	
OTS_FileNode::setChild()	147
<i>Sets first child.</i>	
OTS_FileNode::setNext()	147
<i>Sets next sibling.</i>	
OTS_Group	150
<i>OTServ user group abstraction.</i>	
OTS_Group::count()	151
<i>Returns number of player within.</i>	
OTS_Group::getIterator()	154
<i>Returns players iterator.</i>	
OTS_Group::getMaxDepotItems()	154
<i>Maximum count of items in depot.</i>	
OTS_Group::getMaxVIPList()	155
<i>Maximum count of players in VIP list.</i>	
OTS_Group::getName()	155
<i>Group name.</i>	
OTS_Group::getId()	153
<i>Group ID.</i>	
OTS_Group::getFlags()	153
<i>Rights flags.</i>	
OTS_Group::delete()	152
<i>Deletes group.</i>	
OTS_Group::getAccess()	152
<i>Access level.</i>	
OTS_Group::getCustomField()	152
<i>Reads custom field.</i>	
OTS_FileNode::isValid()	146
<i>Checks if there is anything left in stream.</i>	
OTS_FileNode::getType()	145
<i>Returs node type.</i>	
OTS_FileLoader::loadFile()	139
<i>Opens file.</i>	
OTS_FileLoader::setCacheDriver()	140
<i>Sets cache handler.</i>	
OTS_FileLoader:: clone()	140
<i>Creates clone of object.</i>	
OTS_FileLoader:: set()	141
<i>Magic PHP5 method.</i>	
OTS_FileLoader::\$root	139
<i>Root node.</i>	
OTS_FileLoader::\$cache	139
<i>Cache handler.</i>	
OTS_FileLoader::ESCAPE_CHAR	138

<i>Escape another special byte.</i>	
OTS FileLoader::NODE_END	138
<i>End of node.</i>	
OTS FileLoader::NODE_START	138
<i>Start of node.</i>	
OTS FileLoader::set_state()	141
<i>Magic PHP5 method.</i>	
OTS FileLoader::sleep()	142
<i>Magic PHP5 method.</i>	
OTS FileNode::getNext()	144
<i>Returs next sibling.</i>	
OTS FileNode::getShort()	145
<i>Returns double byte.</i>	
OTS FileNode::getString()	145
<i>Returns string from buffer.</i>	
OTS FileNode::getLong()	144
<i>Returns quater byte.</i>	
OTS FileNode::getChild()	144
<i>Returs first child.</i>	
OTS FileNode	142
<i>OTServ binary file node representation.</i>	
OTS FileNode::getBuffer()	143
<i>Returs properties stream.</i>	
OTS FileNode::getChar()	143
<i>Returns single byte.</i>	
OTS Base_List::rewind()	115
<i>Select rows from database.</i>	
OTS Base_List::resetOrder()	114
<i>Clears ORDER BY clause.</i>	
OTS OTBMFile.php	65
<i>Code in this file bases on oryiginal OTServ OTBM format loading C++ code (iomapotbm.</i>	
OTS Player.php	66
OTS Players_List.php	67
OTS Spell.php	68
OTS MonstersList.php	64
OTS Monster.php	63
OTS ItemsList.php	60
<i>Code in this file bases on oryiginal OTServ items loading C++ code (itemloader.</i>	
OTS ItemType.php	61
<i>Code in this file bases on oryiginal OTServ items loading C++ code (items.</i>	
OTS MapCoords.php	62
<i>Code in this file bases on oryiginal OTServ OTBM format loading C++ code (iomapotbm.</i>	
OTS SpellsList.php	69
OTS SQLField.php	70
OTS Account::block()	92
<i>Blocks account.</i>	
OTS Account::count()	92
<i>Returns number of player within.</i>	
OTS Account::create()	93
<i>Creates new account.</i>	
OTS Account::createEx()	94
<i>Creates new account.</i>	
OTS Account::ban()	91

<i>Bans current account.</i>	
OTS Account	91
<i>OTServ account abstraction.</i>	
OTS SQLFilter.php	71
OTS SQLite Results.php	72
OTS VocationsList.php	73
OTS Item.php	59
OTS InfoRespond.php	58
OTS DB MySQL.php	44
OTS DB ODBC.php	45
OTS DB PostgreSQL.php	46
OTS DB SQLite.php	47
OTS Container.php	43
OTS Base List.php	42
OTS Account.php	39
OTS Accounts List.php	40
OTS Base DAO.php	41
OTS FileLoader.php	48
<i>Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader.</i>	
OTS FileNode.php	49
<i>Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader.</i>	
OTS Guilds List.php	55
OTS House.php	56
OTS HousesList.php	57
OTS GuildRanks List.php	54
OTS GuildRank.php	53
OTS Group.php	50
OTS Groups List.php	51
OTS Guild.php	52
OTS Account::delete()	95
<i>Deletes account.</i>	
OTS Account::find()	95
<i>Loads account by it's e-mail address.</i>	
OTS Base DAO	108
<i>Basic data access object routines.</i>	
OTS Base DAO::\$db	108
<i>Database connection.</i>	
OTS Base DAO:: clone()	109
<i>Creates clone of object.</i>	
OTS Base DAO:: set_state()	109
<i>Magic PHP5 method.</i>	
OTS Accounts List::init()	107
<i>Sets list parameters.</i>	
OTS Accounts List::deleteAccount()	107
<i>Deletes account.</i>	
OTS Account:: get()	105
<i>Magic PHP5 method.</i>	
OTS Account:: set()	106
<i>Magic PHP5 method.</i>	
OTS Accounts List	106
<i>List of accounts.</i>	
OTS Base DAO:: sleep()	110
<i>Magic PHP5 method.</i>	

OTS Base DAO:: wakeup()	110
<i>Magic PHP5 method.</i>	
OTS Base List::init()	113
<i>Sets list parameters.</i>	
OTS Base List::key()	113
<i>Current cursor position.</i>	
OTS Base List::next()	114
<i>Moves to next row.</i>	
OTS Base List::orderBy()	114
<i>Appends sorting rule.</i>	
OTS Base List::current()	112
<i>Returns current row.</i>	
OTS Base List::count()	112
<i>Returns number of rows on list in current criterium.</i>	
OTS Base List	110
<i>Basic list class routines.</i>	
OTS Base List::\$class	111
<i>Class of generated objects.</i>	
OTS Base List::\$table	111
<i>Default table name for queries.</i>	
OTS Account::unblock()	105
<i>Unblocks account.</i>	
OTS Account::unban()	104
<i>Deletes ban from current account.</i>	
OTS Account::getPACCDays()	98
<i>PACC days.</i>	
OTS Account::getPassword()	98
<i>Account's password.</i>	
OTS Account::getPlayers()	99
<i>List of characters on account.</i>	
OTS Account::getPlayersList()	99
<i>List of characters on account.</i>	
OTS Account::getIterator()	97
<i>Returns players iterator.</i>	
OTS Account::getId()	97
<i>Account number.</i>	
OTS Account::getCustomField()	95
<i>Reads custom field.</i>	
OTS Account::getEmail()	96
<i>E-mail address.</i>	
OTS Account::getGroup()	97
<i>Returns group of this account.</i>	
OTS Account::isBanned()	100
<i>Checks if account is banned.</i>	
OTS Account::isBlocked()	100
<i>Checks if account is blocked.</i>	
OTS Account::setGroup()	103
<i>Assigns account to group.</i>	
OTS Account::setPACCDays()	103
<i>Sets PACC days count.</i>	
OTS Account::setPassword()	104
<i>Sets account's password.</i>	
OTS Account::setEmail()	102

<i>Sets account's email.</i>	
OTS Account::setCustomField()	102
<i>Writes custom field.</i>	
OTS Account::isLoading()	100
<i>Checks if object is loaded.</i>	
OTS Account::load()	101
<i>Loads account with given number.</i>	
OTS Account::save()	101
<i>Updates account in database.</i>	
OTS Group::getPlayers()	155
<i>List of characters in given group.</i>	
OTS Group::getPlayersList()	156
<i>List of characters in group.</i>	
OTS InfoRespond::getMapHeight()	207
<i>Returns map height.</i>	
OTS InfoRespond::getMapName()	208
<i>Returns map name.</i>	
OTS InfoRespond::getMapWidth()	208
<i>Returns map width.</i>	
OTS InfoRespond::getMaxPlayers()	208
<i>Returns maximum amount of players online.</i>	
OTS InfoRespond::getMapAuthor()	207
<i>Returns map author.</i>	
OTS InfoRespond::getLocation()	207
<i>Returns server location.</i>	
OTS InfoRespond::getClientVersion()	206
<i>Returns dedicated version of client.</i>	
OTS InfoRespond::getEmail()	206
<i>Returns owner e-mail.</i>	
OTS InfoRespond::getIP()	206
<i>Returns server IP.</i>	
OTS InfoRespond::getMonstersCount()	209
<i>Returns number of all monsters on map.</i>	
OTS InfoRespond::getMOTD()	209
<i>Returns server's Message Of The Day</i>	
OTS InfoRespond::getServer()	211
<i>Returns server attribute.</i>	
OTS InfoRespond::getServerVersion()	211
<i>Returns server version.</i>	
OTS InfoRespond::getTSPQVersion()	211
<i>Returns version of root element.</i>	
OTS InfoRespond::getUptime()	212
<i>Returns server uptime.</i>	
OTS InfoRespond::getPort()	210
<i>Returns server port.</i>	
OTS InfoRespond::getPlayersPeak()	210
<i>Returns record of online players.</i>	
OTS InfoRespond::getName()	209
<i>Returns server name.</i>	
OTS InfoRespond::getOnlinePlayers()	210
<i>Returns current amount of players online.</i>	
OTS InfoRespond::getOwner()	210
<i>Returns owner name.</i>	

OTS InfoRespond	205
<i>Wrapper for 'info' respond's DOMDocument.</i>	
OTS HousesList::set_state()	204
<i>Magic PHP5 method.</i>	
OTS House::setPaid()	197
<i>Sets paid date.</i>	
OTS House::setWarnings()	198
<i>Sets house warnings.</i>	
OTS House::get()	198
<i>Magic PHP5 method.</i>	
OTS House::set()	199
<i>Magic PHP5 method.</i>	
OTS House::setOwner()	197
<i>Sets house owner.</i>	
OTS House::save()	196
<i>Saves info in database.</i>	
OTS House::getTownId()	196
<i>Returns town ID in which house is located.</i>	
OTS House::getTownName()	196
<i>Returns town name.</i>	
OTS House::getWarnings()	196
<i>Returns house warnings.</i>	
OTS House::sleep()	199
<i>Magic PHP5 method.</i>	
OTS HousesList	200
<i>Wrapper for houses list.</i>	
OTS HousesList::offsetGet()	202
<i>Returns item from given position.</i>	
OTS HousesList::offsetSet()	203
<i>This method is implemented for ArrayAccess interface.</i>	
OTS HousesList::offsetUnset()	203
<i>This method is implemented for ArrayAccess interface.</i>	
OTS HousesList::offsetExists()	202
<i>Checks if given element exists.</i>	
OTS HousesList::getIterator()	202
<i>Returns iterator handle for loops.</i>	
OTS HousesList::count()	200
<i>Returns amount of houses.</i>	
OTS HousesList::getHouse()	201
<i>Returns house information.</i>	
OTS HousesList::getHouseId()	201
<i>Returns ID of house with given name.</i>	
OTS InfoRespond::getURL()	212
<i>Returns server website.</i>	
OTS InfoRespond::get()	212
<i>Magic PHP5 method.</i>	
OTS ItemsList::getBuildVersion()	223
<i>Returns build version.</i>	
OTS ItemsList::getClientVersion()	224
<i>Returns client version.</i>	
OTS ItemsList::getItemType()	224
<i>Returns given item type.</i>	
OTS ItemsList::getItemTypeId()	224

<i>Finds item type by it's name.</i>	
OTS_ItemsList::current()	223
<i>Returns item at current position in iterator.</i>	
OTS_ItemsList::count()	223
<i>Returns amount of items loaded.</i>	
OTS_ItemsList::ITEM_ATTR_SPEED	222
<i>Speed.</i>	
OTS_ItemsList::ITEM_ATTR_TOPORDER	222
<i>Always-on-top order.</i>	
OTS_ItemsList::ROOT_ATTR_VERSION	222
<i>Root file attribute.</i>	
OTS_ItemsList::getItemTypesList()	225
<i>Returns all loaded items.</i>	
OTS_ItemsList::getIterator()	225
<i>Returns iterator handle for loops.</i>	
OTS_ItemsList::offsetGet()	228
<i>Returns item from given position.</i>	
OTS_ItemsList::offsetSet()	228
<i>This method is implemented for ArrayAccess interface.</i>	
OTS_ItemsList::offsetUnset()	229
<i>This method is implemented for ArrayAccess interface.</i>	
OTS_ItemsList::rewind()	229
<i>Resets iterator index.</i>	
OTS_ItemsList::offsetExists()	227
<i>Checks if given element exists.</i>	
OTS_ItemsList::next()	227
<i>Moves to next iterator item.</i>	
OTS_ItemsList::getOTBVersion()	226
<i>Returns OTB file version.</i>	
OTS_ItemsList::key()	226
<i>Returns ID of current position.</i>	
OTS_ItemsList::loadItems()	226
<i>Loads items.</i>	
OTS_ItemsList::ITEM_ATTR_SERVERID	221
<i>Server ID.</i>	
OTS_ItemsList::ITEM_ATTR_LIGHT2	221
<i>Light.</i>	
OTS_Item::getItemType()	215
<i>Returns type of item.</i>	
OTS_Item::setAttributes()	216
<i>Sets item attributes.</i>	
OTS_Item::setCount()	216
<i>Sets count of item.</i>	
OTS_Item::get()	217
<i>Magic PHP5 method.</i>	
OTS_Item::getId()	215
<i>Returns item type.</i>	
OTS_Item::getCount()	215
<i>Returns count of item.</i>	
OTS_Item	213
<i>Single item representation.</i>	
OTS_Item::count()	214
<i>Count value for current item.</i>	

OTS_Item::getAttributes()	214
<i>Returns item custom attributes.</i>	
OTS_ItemsList	217
<i>Items list loader.</i>	
OTS_ItemsList::CLIENT_VERSION_750	218
<i>Tibia client 7.</i>	
OTS_ItemsList::CLIENT_VERSION_792	220
<i>Tibia client 7.</i>	
OTS_ItemsList::CLIENT_VERSION_800	220
<i>Tibia client 8.</i>	
OTS_ItemsList::ITEM_ATTR_CLIENTID	221
<i>Client ID.</i>	
OTS_ItemsList::CLIENT_VERSION_790	220
<i>Tibia client 7.</i>	
OTS_ItemsList::CLIENT_VERSION_780	219
<i>Tibia client 7.</i>	
OTS_ItemsList::CLIENT_VERSION_755	218
<i>Tibia client 7.</i>	
OTS_ItemsList::CLIENT_VERSION_760	219
<i>Tibia client 7.</i>	
OTS_ItemsList::CLIENT_VERSION_770	219
<i>Tibia client 7.</i>	
OTS_House::getTiles()	195
<i>Returns tiles list.</i>	
OTS_House::getSize()	195
<i>Returns house size.</i>	
OTS_Guild::getId()	169
<i>Guild ID.</i>	
OTS_Guild::getIterator()	169
<i>Returns ranks iterator.</i>	
OTS_Guild::getName()	170
<i>Guild name.</i>	
OTS_Guild::getOwner()	170
<i>Returns owning player of this player.</i>	
OTS_Guild::getGuildRanksList()	169
<i>List of ranks in guild.</i>	
OTS_Guild::getGuildRanks()	168
<i>Reads all ranks that are in this guild.</i>	
OTS_Guild::find()	167
<i>Loads guild by it's name.</i>	
OTS_Guild::getCreationData()	167
<i>Guild creation data.</i>	
OTS_Guild::getCustomField()	167
<i>Reads custom field.</i>	
OTS_Guild::invite()	171
<i>Invites player to guild.</i>	
OTS_Guild::isLoading()	171
<i>Checks if object is loaded.</i>	
OTS_Guild::setCreationData()	174
<i>Sets guild creation data.</i>	
OTS_Guild::setCustomField()	174
<i>Writes custom field.</i>	
OTS_Guild::setInvitesDriver()	175

<i>Assigns invites handler.</i>	
OTS_Guild::setName()	175
<i>Sets players's name.</i>	
OTS_Guild::save()	173
<i>Saves guild in database.</i>	
OTS_Guild::request()	173
<i>Requests membership in guild for player player.</i>	
OTS_Guild::listInvites()	171
<i>Returns list of invited players.</i>	
OTS_Guild::listRequests()	172
<i>Returns list of players that requested membership.</i>	
OTS_Guild::load()	172
<i>Loads guild with given id.</i>	
OTS_Guild::deleteRequest()	166
<i>Deletes request from player.</i>	
OTS_Guild::deleteInvite()	166
<i>Deletes invitation for player to guild.</i>	
OTS_Group::setFlags()	159
<i>Sets rights flags.</i>	
OTS_Group::setMaxDepotItems()	159
<i>Sets maximum count of items in depot.</i>	
OTS_Group::setMaxVIPList()	160
<i>Sets maximum count of players in VIP list.</i>	
OTS_Group::setName()	160
<i>Sets group's name.</i>	
OTS_Group::setCustomField()	158
<i>Writes custom field.</i>	
OTS_Group::setAccess()	157
<i>Sets access level.</i>	
OTS_Group::isLoading()	156
<i>Checks if object is loaded.</i>	
OTS_Group::load()	157
<i>Loads group with given id.</i>	
OTS_Group::save()	157
<i>Saves account in database.</i>	
OTS_Group::__get()	160
<i>Magic PHP5 method.</i>	
OTS_Group::__set()	161
<i>Magic PHP5 method.</i>	
OTS_Guild::acceptRequest()	164
<i>Accepts player.</i>	
OTS_Guild::count()	165
<i>Returns number of ranks within.</i>	
OTS_Guild::delete()	165
<i>Deletes guild.</i>	
OTS_Guild::acceptInvite()	164
<i>Finalise invitation.</i>	
OTS_Guild	163
<i>OTServ guild abstraction.</i>	
OTS_Groups_List	162
<i>List of groups.</i>	
OTS_Groups_List::deleteGroup()	162
<i>Deletes group.</i>	

OTS_Groups_List::init()	163
<i>Sets list parameters.</i>	
OTS_Guild::setOwner()	176
<i>Assigns guild to owner.</i>	
OTS_Guild::setRequestsDriver()	176
<i>Assigns requests handler.</i>	
OTS_GuildRanks_List::deleteGuildRank()	189
<i>Deletes guild rank.</i>	
OTS_GuildRanks_List::init()	189
<i>Sets list parameters.</i>	
OTS_Guilds_List	190
<i>List of guilds.</i>	
OTS_Guilds_List::deleteGuild()	190
<i>Deletes guild.</i>	
OTS_GuildRanks_List	188
<i>List of guild ranks.</i>	
OTS_GuildRank::set()	188
<i>Magic PHP5 method.</i>	
OTS_GuildRank::setLevel()	186
<i>Sets rank's access level within guild.</i>	
OTS_GuildRank::setName()	187
<i>Sets rank's name.</i>	
OTS_GuildRank::get()	187
<i>Magic PHP5 method.</i>	
OTS_Guilds_List::init()	191
<i>Sets list parameters.</i>	
OTS_House	191
<i>Wrapper for house information.</i>	
OTS_House::getOwner()	194
<i>Returns current house owner.</i>	
OTS_House::getPaid()	194
<i>Returns paid date.</i>	
OTS_House::getRent()	195
<i>Returns house rent cost.</i>	
OTS_House::getName()	194
<i>Return house's name.</i>	
OTS_House::getId()	193
<i>Returns house's ID.</i>	
OTS_House::addTile()	192
<i>Adds tile to house.</i>	
OTS_House::delete()	193
<i>Deletes house info from database.</i>	
OTS_House::getEntry()	193
<i>Returns entry position.</i>	
OTS_GuildRank::setGuild()	186
<i>Assigns rank to guild.</i>	
OTS_GuildRank::setCustomField()	185
<i>Writes custom field.</i>	
OTS_GuildRank::count()	179
<i>Returns number of player within.</i>	
OTS_GuildRank::delete()	179
<i>Deletes guild rank.</i>	
OTS_GuildRank::find()	180

<i>Loads rank by it's name.</i>	
OTS_GuildRank::getCustomField()	180
<i>Reads custom field.</i>	
OTS_GuildRank	179
<i>OTServ guild rank abstraction.</i>	
OTS_Guild::__sleep()	178
<i>Magic PHP5 method.</i>	
OTS_Guild::__clone()	177
<i>Creates clone of object.</i>	
OTS_Guild::__get()	177
<i>Magic PHP5 method.</i>	
OTS_Guild::__set()	177
<i>Magic PHP5 method.</i>	
OTS_GuildRank::getGuild()	181
<i>Returns guild of this rank.</i>	
OTS_GuildRank::getId()	181
<i>Rank ID.</i>	
OTS_GuildRank::isLoading()	184
<i>Checks if object is loaded.</i>	
OTS_GuildRank::load()	184
<i>Loads rank with given id.</i>	
OTS_GuildRank::save()	185
<i>Saves rank in database.</i>	
OTS_GuildRank::getPlayersList()	183
<i>List of characters with current rank.</i>	
OTS_GuildRank::getPlayers()	183
<i>Reads all players who has this rank set.</i>	
OTS_GuildRank::getIterator()	182
<i>Returns players iterator.</i>	
OTS_GuildRank::getLevel()	182
<i>Rank's access level.</i>	
OTS_GuildRank::getName()	183
<i>Rank name.</i>	
OTS.php	38
<i>This file contains main toolkit class.</i>	

P

POT::getItemsList()	399
<i>Returns list of laoded items.</i>	
POT::getInstantsList()	398
<i>Returns list of instants.</i>	
POT::getInstant()	398
<i>Returns given instant spell.</i>	
POT::getInstance()	398
<i>Singleton.</i>	
POT::getMap()	399
<i>Returns loaded map.</i>	
POT::getMonster()	400
<i>Returns loaded data of given monster.</i>	
POT::getSpellsList()	402
<i>Returns list of laoded spells.</i>	

POT::getRunesList()	401
<i>Returns list of runes.</i>	
POT::getRune()	401
<i>Returns given rune spell.</i>	
POT::getMonstersList()	400
<i>Returns list of loaded monsters.</i>	
POT::getHousesList()	397
<i>Returns list of loaded houses.</i>	
POT::getDBHandle()	397
<i>Returns database connection handle.</i>	
POT::banIP()	393
<i>Bans given IP number.</i>	
POT::areVocationsLoaded()	392
<i>Checks if vocations are loaded.</i>	
POT::areSpellsLoaded()	392
<i>Checks if spells are loaded.</i>	
POT::areMonstersLoaded()	391
<i>Checks if monsters are loaded.</i>	
POT::connect()	393
<i>Connects to database.</i>	
POT::createFilter()	395
<i>Creates lists filter.</i>	
POT::getConjuresList()	396
<i>Returns list of conjure spells.</i>	
POT::getConjure()	395
<i>Returns given conjure spell.</i>	
POT::createObject()	395
<i>Creates OTServ DAO class instance.</i>	
POT::getVocationId()	402
<i>Returns vocation's ID.</i>	
POT::getVocationName()	403
<i>Returns name of given vocation's ID.</i>	
POT::unbanIP()	411
<i>Deletes ban from given IP number.</i>	
POT::setPOTPath()	411
<i>Set POT directory.</i>	
POT::setMapCache()	410
<i>Presets cache handler for OTBM loader.</i>	
POT::setItemsCache()	410
<i>Presets cache handler for items loader.</i>	
POT::unloadHouses()	412
<i>Unloads houses list.</i>	
POT::unloadItems()	412
<i>Unloads items list.</i>	
POT::unloadVocations()	414
<i>Unloads vocations list.</i>	
POT::unloadSpells()	414
<i>Unloads spells list.</i>	
POT::unloadMonsters()	413
<i>Unloads monsters list.</i>	
POT::unloadMap()	413
<i>Unloads OTBM map.</i>	
POT::serverStatus()	409

<u>Queries server status.</u>	408
<u>POT::loadVocations()</u>	408
<i>Loads vocations list.</i>	
<u>POT::loadClass()</u>	405
<i>Loads POT class file.</i>	
<u>POT::isMapLoaded()</u>	404
<i>Checks if OTBM is loaded.</i>	
<u>POT::isIPBanned()</u>	404
<i>Checks if given IP is banned.</i>	
<u>POT::getVocationsList()</u>	403
<i>Returns vocations list object.</i>	
<u>POT::loadHouses()</u>	405
<i>Loads houses list file.</i>	
<u>POT::loadItems()</u>	406
<i>Loads items list.</i>	
<u>POT::loadSpells()</u>	407
<i>Loads spells list.</i>	
<u>POT::loadMonsters()</u>	407
<i>Loads monsters mapping file.</i>	
<u>POT::loadMap()</u>	406
<i>Loads OTBM map.</i>	
<u>POT::areItemsLoaded()</u>	391
<i>Checks if items are loaded.</i>	
<u>POT::areHousesLoaded()</u>	391
<i>Checks if houses are loaded.</i>	
<u>POT::DIRECTION WEST</u>	379
<i>West.</i>	
<u>POT::DIRECTION SOUTH</u>	378
<i>South.</i>	
<u>POT::DIRECTION NORTH</u>	378
<i>North.</i>	
<u>POT::DIRECTION EAST</u>	378
<i>East.</i>	
<u>POT::ORDER ASC</u>	379
<i>Ascencind sorting order.</i>	
<u>POT::ORDER DESC</u>	379
<i>Descending sorting order.</i>	
<u>POT::SKILL CLUB</u>	381
<i>Club fighting.</i>	
<u>POT::SKILL AXE</u>	381
<i>Axe fighting.</i>	
<u>POT::SEX MALE</u>	380
<i>Male gender.</i>	
<u>POT::SEX FEMALE</u>	380
<i>Female gender.</i>	
<u>POT::DEPOT_SID FIRST</u>	377
<i>First depot item sid.</i>	
<u>POT::DB SQLITE</u>	377
<i>SQLite driver.</i>	
<u>POT::BAN ACCOUNT</u>	375
<i>Account ban.</i>	
<u>POT</u>	374
<i>Main POT class.</i>	

POT class preview	5
PHP 5.0	3
POT::BAN_IP	375
<i>IP ban.</i>	
POT::BAN_PLAYER	375
<i>Player ban.</i>	
POT::DB_PGSQL	377
<i>PostgreSQL driver.</i>	
POT::DB_ODBC	376
<i>ODBC driver.</i>	
POT::DB_MYSQL	376
<i>MySQL driver.</i>	
POT::SKILL_DISTANCE	381
<i>Distance fighting.</i>	
POT::SKILL_FISHING	382
<i>Fishing.</i>	
POT::SPELL_INSTANT	388
<i>Instant spell.</i>	
POT::SPELL_CONJURE	388
<i>Conjure spell.</i>	
POT::SLOT_RING	387
<i>Ring slot.</i>	
POT::SLOT_RIGHT	387
<i>Right hand slot.</i>	
POT::SPELL_RUNE	388
<i>Rune spell.</i>	
POT::VOCATION_DRUID	389
<i>Druid.</i>	
POT::VOCATION_SORCERER	390
<i>Sorcerer.</i>	
POT::VOCATION_PALADIN	390
<i>Paladin.</i>	
POT::VOCATION_NONE	390
<i>None vocation.</i>	
POT::VOCATION_KNIGHT	389
<i>Knight.</i>	
POT::SLOT_NECKLACE	386
<i>Necklace slot.</i>	
POT::SLOT_LEGS	386
<i>Legs slot.</i>	
POT::SLOT_AMMO	383
<i>Ammunition slot.</i>	
POT::SKILL_SWORD	383
<i>Sword fighting.</i>	
POT::SKILL_SHIELDING	383
<i>Shielding.</i>	
POT::SKILL_FIST	382
<i>Fist fighting.</i>	
POT::SLOT_ARMOR	384
<i>Armor slot.</i>	
POT::SLOT_BACKPACK	384
<i>Backpack slot.</i>	
POT::SLOT_LEFT	385

	<i>Left hand slot.</i>	
POT::SLOT_HEAD		385
	<i>Head slot.</i>	
POT::SLOT_FEET		385
	<i>Boots slot.</i>	
POT		1

Q	
Quick start	6

R	
README	429
RULES	428

S	
Server online status	19