

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

OTS Item.php	55
OTS ItemsList.php	56
OTS ItemType.php	57
OTS MapCoords.php	58
OTS Monster.php	59
OTS OTBMFile.php	60
OTS Player.php	61
OTS Players List.php	62
OTS Spell.php	63
OTS SQLField.php	64
OTS SQLFilter.php	65
OTS SQLite Results.php	66
Package POT Classes	67
Class E OTS ErrorCode	67
Constructor __construct	67
Class E OTS FileLoaderError	68
Class Constant ERROR_CAN_NOT_OPEN	68
Class Constant ERROR_EOF	68
Class Constant ERROR_INVALID_FILE_VERSION	69
Class Constant ERROR_INVALID_FORMAT	69
Class Constant ERROR_NOT_OPEN	69
Class Constant ERROR_SEEK_ERROR	70
Class Constant ERROR_TELL_ERROR	70
Class E OTS NoDriver	71
Class E OTS NotLoaded	71
Class E OTS OTBMErrors	72
Class Constant LOADMAPERROR_OUTDATEDHEADER	72
Class Constant LOADMAPERROR_UNKNOWNNODETYPE	72
Class E OTS OutOfBuffer	73
Class IOTS DAO	73
Constructor __construct	73
Class IOTS DB	74
Constructor __construct	74
Method fieldName	75
Method lastInsertId	75
Method limit	76
Method SQLquery	76
Method SQLquote	77
Method tableName	77
Class IOTS FileCache	78
Method readCache	78
Method writeCache	78
Class IOTS GuildAction	79
Constructor __construct	80
Method addRequest	80
Method deleteRequest	81
Method listRequests	81
Method submitRequest	81
Class IOTS ItemsCache	82

Method readItems	82
Method writeItems	83
Class OTS Account	83
Method ban	84
Method block	84
Method count	85
Method create	85
example: account.php	85
Method createEx	86
Method delete	87
Method find	87
Method getCustomField	88
Method getEmail	88
Method getGroup	89
Method getId	89
Method getIterator	90
Method getPACCDays	90
Method getPassword	91
Method getPlayers	91
Method getPlayersList	91
Method isBanned	92
Method isBlocked	92
Method isLoaded	93
Method load	93
Method save	93
Method setCustomField	94
Method setEmail	95
Method setGroup	95
Method setPACCDays	96
Method setPassword	96
Method unban	97
Method unblock	97
Class OTS Accounts List	97
Method deleteAccount	98
Method init	98
Class OTS Base DAO	99
Var \$db	99
Constructor construct	100
Method clone	100
Method set_state	100
Method sleep	101
Method wakeup	101
Class OTS Base List	102
Var \$class	102
Var \$table	102
Constructor construct	103
Method count	103
Method current	104
Method init	104

Method key	104
Method next	105
Method orderBy	105
Method resetOrder	105
Method rewind	106
Method setFilter	106
Method setLimit	107
Method setOffset	107
Method valid	107
Method set state	108
Method sleep	108
Method wakeup	109
Class OTS Container	109
Method addItem	109
Method count	110
Method current	110
Method key	111
Method next	111
Method removeItem	111
Method rewind	112
Method valid	112
Class OTS DB MySQL	113
Constructor construct	113
Method fieldName	114
Method limit	114
Method SQLquery	115
Method SQLquote	115
Method tableName	116
Class OTS DB ODBC	116
Constructor construct	117
Method fieldName	117
Method limit	118
Method SQLquery	118
Method SQLquote	119
Method tableName	119
Class OTS DB PostgreSQL	120
Constructor construct	120
Method fieldName	121
Method limit	121
Method SQLquery	122
Method SQLquote	122
Method tableName	123
Class OTS DB SQLite	124
Constructor construct	124
Method fieldName	124
Method limit	125
Method SQLquery	125
Method SQLquote	126
Method tableName	126

Class OTS_FileLoader	127
Class Constant ESCAPE_CHAR	127
Class Constant NODE_END	128
Class Constant NODE_START	128
Var \$cache	128
Var \$root	129
Method loadFile	129
Method setCacheDriver	130
Method clone	130
Method set_state	131
Method sleep	131
Class OTS_FileNode	132
Method getBuffer	132
Method getChar	132
Method getChild	133
Method getLong	133
Method getNext	133
Method getShort	134
Method getString	134
Method getType	134
Method isValid	135
Method setBuffer	135
Method setChild	135
Method setNext	136
Method setType	136
Method skip	137
Method clone	137
Method set_state	138
Class OTS_Group	138
Method count	139
Method delete	139
Method getAccess	139
Method getCustomField	140
Method getFlags	140
Method getId	141
Method getIterator	141
Method getMaxDepotItems	142
Method getMaxVIPList	142
Method getName	142
Method getPlayers	143
Method getPlayersList	143
Method isLoaded	144
Method load	144
Method save	145
Method setAccess	145
Method setCustomField	145
Method setFlags	146
Method setMaxDepotItems	147
Method setMaxVIPList	147

Method setName	148
Class OTS_Groups_List	148
Method deleteGroup	148
Method init	149
Class OTS_Guild	149
Method acceptInvite	150
Method acceptRequest	150
Method count	151
Method delete	151
Method deleteInvite	152
Method deleteRequest	152
Method find	153
Method getCreationData	153
Method getCustomField	154
Method getGuildRanks	154
Method getGuildRanksList	155
Method getId	155
Method getIterator	155
Method getName	156
Method getOwner	156
Method invite	157
Method isLoading	157
Method listInvites	157
Method listRequests	158
Method load	158
Method request	159
Method save	159
Method setCreationData	160
Method setCustomField	160
Method setInvitesDriver	161
Method setName	161
Method setOwner	162
Method setRequestsDriver	162
Method clone	163
Method sleep	163
Class OTS_GuildRank	163
Method count	164
Method delete	164
Method find	165
Method getCustomField	165
Method getGuild	166
Method getId	166
Method getIterator	166
Method getLevel	167
Method getName	167
Method getPlayers	168
Method getPlayersList	168
Method isLoading	169
Method load	169

Method save	169
Method setCustomField	170
Method setGuild	170
Method setLevel	171
Method setName	171
Class OTS_GuildRanks_List	172
Method deleteGuildRank	172
Method init	173
Class OTS_Guilds_List	173
Method deleteGuild	174
Method init	174
Class OTS_InfoRespond	175
Method getClientVersion	175
Method getEmail	175
Method getIP	176
Method getLocation	176
Method getMapAuthor	176
Method getMapHeight	177
Method getMapName	177
Method getMapWidth	177
Method getMaxPlayers	178
Method getMonstersCount	178
Method getMOTD	178
Method getName	178
Method getOnlinePlayers	179
Method getOwner	179
Method getPlayersPeak	179
Method getPort	180
Method getServer	180
Method getServerVersion	180
Method getTSPQVersion	181
Method getUptime	181
Method getURL	181
Class OTS_Item	182
Constructor __construct	182
Method count	183
Method getAttributes	183
Method getCount	183
Method getId	184
Method setAttributes	184
Method setCount	184
Class OTS_ItemsList	185
Class Constant CLIENT_VERSION_750	185
Class Constant CLIENT_VERSION_755	186
Class Constant CLIENT_VERSION_760	186
Class Constant CLIENT_VERSION_770	186
Class Constant CLIENT_VERSION_780	187
Class Constant CLIENT_VERSION_790	187
Class Constant CLIENT_VERSION_792	187

Class Constant CLIENT_VERSION_800	188
Class Constant ITEM_ATTR_CLIENTID	188
Class Constant ITEM_ATTR_LIGHT2	188
Class Constant ITEM_ATTR_SERVERID	189
Class Constant ITEM_ATTR_SPEED	189
Class Constant ITEM_ATTR_TOPORDER	189
Class Constant ROOT_ATTR_VERSION	190
Method count	190
Method current	190
Method getBuildVersion	191
Method getClientVersion	191
Method getItemType	191
Method getItemTypeId	192
Method getItemTypesList	192
Method getOTBVersion	193
Method key	193
Method loadItems	193
Method next	194
Method rewind	194
Method valid	194
Method set_state	195
Method wakeup	195
Class OTS_ItemType	196
Class Constant FLAG_ALLOWDISTREAD	196
Class Constant FLAG_ALWAYSONTOP	196
Class Constant FLAG_BLOCK_PATHFIND	197
Class Constant FLAG_BLOCK_PROJECTILE	197
Class Constant FLAG_BLOCK_SOLID	197
Class Constant FLAG_CANNOTDECAY	198
Class Constant FLAG_FLOORCHANGEDOWN	198
Class Constant FLAG_FLOORCHANGEEST	199
Class Constant FLAG_FLOORCHANGENORTH	199
Class Constant FLAG_FLOORCHANGESOUTH	199
Class Constant FLAG_FLOORCHANGEWEST	200
Class Constant FLAG_HANGABLE	200
Class Constant FLAG_HAS_HEIGHT	200
Class Constant FLAG_HORIZONTAL	201
Class Constant FLAG_MOVEABLE	201
Class Constant FLAG_PICKUPABLE	201
Class Constant FLAG_READABLE	202
Class Constant FLAG_ROTABLE	202
Class Constant FLAG_STACKABLE	202
Class Constant FLAG_USEABLE	203
Class Constant FLAG_VERTICAL	203
Class Constant ITEM_GROUP_AMMUNITION	203
Class Constant ITEM_GROUP_ARMOR	204
Class Constant ITEM_GROUP_CONTAINER	204
Class Constant ITEM_GROUP_DOOR	204
Class Constant ITEM_GROUP_FLUID	205

Class Constant ITEM_GROUP_GROUND	205
Class Constant ITEM_GROUP_KEY	205
Class Constant ITEM_GROUP_MAGICFIELD	206
Class Constant ITEM_GROUP_NONE	206
Class Constant ITEM_GROUP_RUNE	206
Class Constant ITEM_GROUP_SPLASH	207
Class Constant ITEM_GROUP_TELEPORT	207
Class Constant ITEM_GROUP_WEAPON	207
Class Constant ITEM_GROUP_WRITEABLE	208
Class Constant ITEM_TYPE_CONTAINER	208
Class Constant ITEM_TYPE_DEPOT	208
Class Constant ITEM_TYPE_DOOR	209
Class Constant ITEM_TYPE_MAGICFIELD	209
Class Constant ITEM_TYPE_MAILBOX	209
Class Constant ITEM_TYPE_NONE	210
Class Constant ITEM_TYPE_TRASHHOLDER	210
Constructor construct	210
Method createItem	211
Method getAttribute	211
Method getAttributesList	212
Method getClientId	212
Method getGroup	212
Method getId	213
Method getName	213
Method getType	213
Method hasHeight	214
Method isAlwaysOnTop	214
Method isBlocking	214
Method isHangable	215
Method isHorizontal	215
Method isMovable	215
Method isPickupable	215
Method isReadable	216
Method isRotable	216
Method isStackable	216
Method isUsable	217
Method isVertical	217
Method setAttribute	217
Method setClientId	218
Method setFlags	218
Method setGroup	219
Method setName	219
Method setType	220
Method set_state	220
Class OTS_MapCoords	221
Constructor construct	221
Method getX	222
Method getY	222
Method getZ	222

Method set state	223
Class OTS Monster	223
Method getArmor	224
Method getAttacks	224
Method getDefense	224
Method getDefenses	225
Method getExperience	225
Method getFlag	225
Method getFlags	226
Method getHealth	226
Method getImmunities	226
Method getLoot	227
Method getManaCost	227
Method getName	227
Method getRace	228
Method getSpeed	228
Method getVoices	228
Method hasImmunity	229
Class OTS OTBMFile	229
Class Constant OTBM ATTR ACTION_ID	230
Class Constant OTBM ATTR DEPOT_ID	230
Class Constant OTBM ATTR DESC	230
Class Constant OTBM ATTR DESCRIPTION	231
Class Constant OTBM ATTR EXT_FILE	231
Class Constant OTBM ATTR EXT_HOUSE_FILE	231
Class Constant OTBM ATTR EXT_SPAWN_FILE	232
Class Constant OTBM ATTR HOUSEDORID	232
Class Constant OTBM ATTR ITEM	232
Class Constant OTBM ATTR RUNE_CHARGES	233
Class Constant OTBM ATTR TELE_DEST	233
Class Constant OTBM ATTR TEXT	233
Class Constant OTBM ATTR TILE_FLAGS	234
Class Constant OTBM ATTR UNIQUE_ID	234
Class Constant OTBM NODE HOUSEDORID	234
Class Constant OTBM NODE ITEM	235
Class Constant OTBM NODE ITEM_DEF	235
Class Constant OTBM NODE MAP_DATA	235
Class Constant OTBM NODE MONSTER	236
Class Constant OTBM NODE ROOTV1	236
Class Constant OTBM NODE SPAWNS	236
Class Constant OTBM NODE SPAWN_AREA	237
Class Constant OTBM NODE TILE	237
Class Constant OTBM NODE TILE_AREA	237
Class Constant OTBM NODE TILE_REF	238
Class Constant OTBM NODE TILE_SQUARE	238
Class Constant OTBM NODE TOWN	238
Class Constant OTBM NODE TOWNS	239
Method count	239
Method current	239

Method getDescription	240
Method getHeight	240
Method getTownID	241
Method getTownName	241
Method getTownsList	241
Method getTownTemple	242
Method getWidth	242
Method key	243
Method loadFile	243
Method next	243
Method rewind	244
Method valid	244
Method set state	245
Method wakeup	245
Class OTS Player	246
Method ban	246
Method delete	246
Method find	247
Method getAccount	247
Method getCap	248
Method getConditions	248
Method getCustomField	249
Method getDepot	249
Method getDirection	250
Method getExperience	250
Method getGroup	251
Method getGuildNick	251
Method getHealth	251
Method getHealthMax	252
Method getId	252
Method getLastIP	253
Method getLastLogin	253
Method getLevel	253
Method getLookAddons	254
Method getLookBody	254
Method getLookFeet	254
Method getLookHead	255
Method getLookLegs	255
Method getLookType	256
Method getLossExperience	256
Method getLossMana	256
Method getLossSkills	257
Method getMagLevel	257
Method getMana	258
Method getManaMax	258
Method getManaSpent	258
Method getName	259
Method getPosX	259
Method getPosY	260

Method getPosZ	260
Method getPremiumEnd	260
Method getRank	261
Method getRankId	261
Method getRedSkullTime	262
Method getSave	262
Method getSex	262
Method getSkill	263
Method getSkillTries	263
Method getSlot	264
Method getSoul	265
Method getTownId	265
Method getVocation	265
Method getVocationName	266
Method hasRedSkull	266
Method isBanned	267
Method isLoaded	267
Method isSaveSet	267
Method load	268
Method save	268
Method setAccount	268
Method setCap	269
Method setConditions	269
Method setCustomField	270
Method setDepot	271
Method setDirection	271
Method setExperience	272
Method setGroup	272
Method setGuildNick	273
Method setHealth	273
Method setHealthMax	274
Method setLastIP	274
Method setLastLogin	274
Method setLevel	275
Method setLookAddons	275
Method setLookBody	276
Method setLookFeet	276
Method setLookHead	277
Method setLookLegs	277
Method setLookType	278
Method setLossExperience	278
Method setLossMana	279
Method setLossSkills	279
Method setMagLevel	279
Method setMana	280
Method setManaMax	280
Method setManaSpent	281
Method setName	281
Method setPosX	282

Method setPosY	282
Method setPosZ	283
Method setPremiumEnd	283
Method setRank	284
Method setRankId	284
Method setRedSkull	284
Method setRedSkullTime	285
Method setSave	285
Method setSex	286
Method setSkill	286
Method setSkillTries	287
Method setSlot	287
Method setSoul	288
Method setTownId	288
Method setVocation	289
Method unban	289
Method unsetRedSkull	290
Method unsetSave	290
Method sleep	290
Class OTS Players List	291
Method deletePlayer	291
Method init	292
Class OTS Spell	292
Constructor construct	293
Method getCharges	293
Method getConjureCount	293
Method getConjureId	294
Method getID	294
Method getLevel	294
Method getMagicLevel	295
Method getMana	295
Method getName	295
Method getReagentId	296
Method getSoul	296
Method getType	296
Method getVocations	297
Method getWords	297
Method hasParams	297
Method isAggresive	298
Method isEnabled	298
Method isFarUseAllowed	298
Method isLearnNeeded	299
Method isPremium	299
Class OTS SQLField	299
Constructor construct	300
Method getName	300
Method getTable	300
Class OTS SQLFilter	301
Class Constant CRITERIUM AND	301

Class Constant CRITERIUM_OR	302
Class Constant OPERATOR_EQUAL	302
Class Constant OPERATOR_GREATER	302
Class Constant OPERATOR_LIKE	303
Class Constant OPERATOR_LOWER	303
Class Constant OPERATOR_NEQUAL	303
Class Constant OPERATOR_NGREATER	304
Class Constant OPERATOR_NLIKE	304
Class Constant OPERATOR_NLOWER	304
Method addFilter	305
Method compareField	305
Method getTables	306
Method sleep	306
Method toString	307
Class POT	307
Class Constant BAN_ACCOUNT	307
Class Constant BAN_IP	308
Class Constant BAN_PLAYER	308
Class Constant DB_MYSQL	309
Class Constant DB_ODBC	309
Class Constant DB_PGSQL	309
Class Constant DB_SQLITE	310
Class Constant DEPOT_SID_FIRST	310
Class Constant DIRECTION_EAST	311
Class Constant DIRECTION_NORTH	311
Class Constant DIRECTION_SOUTH	311
Class Constant DIRECTION_WEST	312
Class Constant ORDER_ASC	312
Class Constant ORDER_DESC	312
Class Constant SEX_FEMALE	313
Class Constant SEX_MALE	313
Class Constant SKILL_AXE	313
Class Constant SKILL_CLUB	314
Class Constant SKILL_DISTANCE	314
Class Constant SKILL_FISHING	315
Class Constant SKILL_FIST	315
Class Constant SKILL_SHIELDING	315
Class Constant SKILL_SWORD	316
Class Constant SLOT_AMMO	316
Class Constant SLOT_ARMOR	317
Class Constant SLOT_BACKPACK	317
Class Constant SLOT_FEET	318
Class Constant SLOT_HEAD	318
Class Constant SLOT_LEFT	318
Class Constant SLOT_LEGS	319
Class Constant SLOT_NECKLACE	319
Class Constant SLOT_RIGHT	320
Class Constant SLOT_RING	320
Class Constant SPELL_CONJURE	320

Class Constant SPELL_INSTANT	321
Class Constant SPELL_RUNE	321
Class Constant VOCATION_DRUID	322
Class Constant VOCATION_KNIGHT	322
Class Constant VOCATION_NONE	322
Class Constant VOCATION_PALADIN	323
Class Constant VOCATION_SORCERER	323
Method banIP	323
Method connect	324
example: connect.php	324
Method createFilter	325
Method createObject	326
Method getConjure	326
Method getConjuresList	327
Method getDBHandle	327
Method getInstance	328
Method getInstant	328
Method getInstantsList	329
Method getMonster	329
Method getMonstersList	329
Method getRune	330
Method getRunesList	330
Method getVocationID	331
Method getVocationName	331
Method getVocationsList	332
Method isIPBanned	332
Method loadClass	333
Method loadMonsters	333
Method loadSpells	334
Method loadVocations	334
Method serverStatus	335
example: example	335
Method setPOTPath	336
example: fakeroot.php	336
Method unbanIP	337
compat.php	338
Appendices	339
Appendix A - Class Trees	340
POT	340
Appendix B - README/CHANGELOG/INSTALL	345
LICENSE	346
CHANGELOG	348
RULES	350
README	351
INSTALL	352
NEWS	353
Appendix D - Todo List	354

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= POT::getInstance()-> createObject('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= $ots->    createObject('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= $ots->    createObject('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= $ots> createObject('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= $ots-> createObject('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 // creates new OTS_Player object
15 $player= $ots-> createObject('Player');
16 $player-> find('Wrzasq');
17
18 /*
19  Items loading example.
20 */
21
22 // loading item from ammunition slot
23 $item= $player-> getSlot(POT::SLOT_AMMO);
24
25 echo $player-> getName(), ' has item with id ', $item-> getId(), ' in his/her ammo slot.', "\n" ;
26
27 // checks if item is a container
28 if($item instanceof OTS_Container)
29 {
30     // list backpack content
31     foreach($item as $inside)
32     {
33         echo 'Container contains item with id ', $inside-> getId(), ' ', "\n" ;
34     }
35 }
36
37 /*
38  Items tree composing example.
39 */
40
41 // creates container - here it would be a depot locker (we pass ID of item to create)
42 $container= new OTS_Container(2590);
43
44 // now let's create depot chest
45 $chest= new OTS_Container(2594);
46
47 // let's put chest inside locker
48 $container-> addItem($chest);
49
50 // now let's put something deeper - into the chest
51 $item1= new OTS_Item(3015);
52 $chest-> addItem($item1);
53
54 // and more...
55 $item2= new OTS_Item(3013);
56 $chest-> addItem($item2);
57
58 // let's set count for an item
59 $item2-> setCount(2);
60
61 /*
62  Here is a tree of items which we created:
63

```

```

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

```

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

Guilds

Guilds system basics.

Basics

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 through guild ranks. To list guild members you first need to list its ranks. Here is an example solution to list members in original 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= $ots->createObject('Guild');
16 $guild->load(1);
17
18 $color= '#FFFFCC';
19
20 echo '<h1>Members of ' . $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( $guild->getGuildRanks() as $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( $guildRank> getPlayers() as $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(as $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= $ots> createObject('Player');
85 $player> load(1);
86

```



```
87 // loads guild with ID 1
88 $guild= $ots->    createObject('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= $ots->createObject('Account');
19 $account->load($number);
20
21 // number is busy
22 if( $account->isLoaded() )
23 {
24     echo 'Account number ', $number, 'is used.', "\n" ;
25 }
26 // it is not
27 else
28 {
29     // generate number from exactly $number - $number range
30     $number= $account->create($number, $number);
31     echo 'Your account number is: ', $number, "\n" ;
32 }
33
34 ?>
```

Server online status

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

Such a simple way

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

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

DOM way

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

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

About OTServ AAC scripts

This small article describes general info about OTServ AAC scripts.

Basics

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

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

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

What is AAC?

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

Types of AAC

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

Why not ingame

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

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

Website AAC HOWTO

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

Main features

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

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

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

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

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

Don't touch!

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

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

Important notes

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

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

Ready scripts

Here are links to some major AAC scripts:

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

Link



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

You can use following code for that:

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


Package POT Procedural Elements

E_OTS_ErrorCode.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.0.6
- **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_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](#)

IOTS_DAO.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](#)

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** 0.0.7
- **Copyright** 2007 (C) by Wrzasq
- **TODO** 0.1.0: Implement `__get()/__set()/__call()/__toString()`; `ArrayAccess` interface.
- **TODO** 0.1.0: Handle instances of loaded data/ directory resources to allow exchange data between classes and objects.
- **TODO** 0.1.0: Put items map handling into POT class or separate vocations, monsters and spells too.
- **TODO** 0.1.0: Get rid of `POT::getInstance()->create*()` calls - use `POT::getInstance()->getDBHandle()` in constructors.
- **TODO** 0.0.9: Houses support.
- **TODO** 1.0.0: Complete phpUnit test.
- **TODO** 0.1.0: Provide suite for automatic items, spells, monsters and other data/ directory info loading.
- **TODO** 1.0.0: SOAP interface for remote controll.
- **TODO** 1.0.0: PHAR and PHK packages.
- **TODO** 1.0.0: More detailed documentation and tutorials, also update examples and tutorials.
- **TODO** 1.0.0: OTAdmin protocol.
- **TODO** 1.0.0: Main POT class as database instance.
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Account.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_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** 0.0.6
- **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** 0.0.7
- **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** 0.0.3
- **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** 0.0.6
- **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** 0.0.6
- **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** 0.0.5
- **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** 0.0.8
- **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** 0.0.8
- **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_InfoRespond.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **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** 0.0.3
- **Copyright** 2007 (C) by Wrzasq
- **TODO** 0.1.0: Provide item type fetching.
- **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** 0.0.8
- **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** 0.0.8
- **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** 0.0.6
- **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** 0.0.6
- **Copyright** 2007 (C) by Wrzasq
- **TODO** 0.1.0: Fetching loot as item types.
- **Since** 0.0.6
- **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
- **Copyright** 2007 (C) by Wrzasq
- **TODO** 0.1.0: Houses support.
- **TODO** 1.0.0: Complete OTBM support: link tiles with items, spawns and houses.
- **TODO** 1.0.0: Spawns support.
- **Since** 0.0.6
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Player.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.0.1
- **Version** 0.0.7
- **Copyright** 2007 (C) by Wrzasq
- **TODO** 0.1.0: Check item types if they are containers during loading slots/depots.
- **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** 0.0.7
- **Copyright** 2007 (C) by Wrzasq
- **TODO** 0.1.0: Link conjures with item types when items support will be added.
- **Since** 0.0.7
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_SQLField.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **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** 0.0.5
- **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](#)

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_NotLoaded

[line 20]

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

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

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

Class E_OTS_OTBMEError

[line 22]

OTBM map loading error.
OTBM map loading error.

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

E_OTS_OTBMErrror::LOADMAPERROR_OUTDATEDHEADER

= 3 *[line 27]*

Unsupported file version.
Unsupported file version.

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

E_OTS_OTBMErrror::LOADMAPERROR_UNKNOWNNODETYPE

= 8 *[line 31]*

Unknown node type.
Unknown node type.

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

Class E_OTS_OutOfBuffer

[line 22]

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

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

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

Class IOTS_DAO

[line 22]

OTServ database object.

OTServ database object.

This interface indicates that class is a OTServ DAO class.

- **Package** POT
- **Version** 0.0.5
- **Since** 0.0.1

Constructor *void* function IOTS_DAO::__construct(\$db) [line 30]

Function Parameters:

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

DAO objects must be initialized with a database.

DAO objects must be initialized with a database.

- **Version** 0.0.5
- **Deprecated** 0.0.5 This constructor convention won't be part of interface in future.
- **Since** 0.0.1
- **Access** public

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 objects 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 recommended that your implementations calls assignment functions of \$guild to automatically 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 22*]

OTServ account abstraction.

OTServ account abstraction.

- **Package** POT

- **Version** 0.0.6
- **Version** 0.0.1
- **Since** 0.0.1

void function OTS_Account::ban([\$time = 0]) [line 464]

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 310]

Blocks account.

Blocks account.

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

int function `OTS_Account::count()` [*line 555*]

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 47*]

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 = $ots->createObject('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 115*]

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 518]

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 139]

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 357*]

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_OTN_NotLoaded If account is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

string function OTS_Account::getEmail() [*line 262*]

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 204*]

Returns group of this account.

Returns group of this account.

- **Version** 0.0.6
- **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 185*]

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 542*]

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 323*]

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 235*]

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 407*]

List of characters on account.

List of characters on account.

- **Version** 0.0.5
- **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 437*]

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

bool function OTS_Account::isBanned() [*line 499*]

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 289*]

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 156*]

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 126*]

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 167*]

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 383]

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 277]

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 224]

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 340]

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 250]

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 481]

Deletes ban from current account.

Deletes ban from current account.

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

void function OTS_Account::unblock() [line 302]

Unblocks account.

Unblocks account.

- **Version** 0.0.1
- **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** 0.0.6
- **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(\$db) [line 36]

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

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** 0.0.6
- **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 22]

Basic list class routines.

Basic list class routines.

- **Package** POT
- **Version** 0.0.7
- **Version** 0.0.5
- **Abstract Element**
- **Since** 0.0.5

OTS_Base_List::\$class

string = [line 78]

Class of generated objects.

Class of generated objects.

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

OTS_Base_List::\$table

string = [line 71]

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 85*]

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 240*]

Returns number of accounts on list in current criterium.

Returns number of accounts on list in current criterium.

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

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

LOTS_DAO function OTS_Base_List::current() [*line 189*]

Returns current row.

Returns current row.

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

void function OTS_Base_List::init() [*line 94*]

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 219*]

Current cursor position.

Current cursor position.

- **Version** 0.0.5

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

void function OTS_Base_List::next() [line 209]

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 273]

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 261]

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 201]

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 253]

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 155]

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 172]

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 229]

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_state(\$properties) [*line 130*]

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 104*]

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 116*]

Magic PHP5 method.

Magic PHP5 method.
Allows object unserialisation.

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

Class OTS_Container

[*line 20*]

Container item representation.

Container item representation.

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

void function OTS_Container::addItem(\$item) [*line 34*]

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 65*]

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 75*]

Returns current item.

Returns current item.

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

mixed function OTS_Container::key() [*line 93*]

Current cursor position.

Current cursor position.

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

void function OTS_Container::next() [*line 83*]

Moves to next item.

Moves to next item.

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

void function OTS_Container::removeItem(\$item) [*line 46*]

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 111*]

Resets internal items array pointer.

Resets internal items array pointer.

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

bool function OTS_Container::valid() [*line 103*]

Checks if there are any items left.

Checks if there are any items left.

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

Class OTS_DB_MySQL

[line 22]

MySQL connection interface.

MySQL connection interface.

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

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

Function Parameters:

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

Creates database connection.

Creates database connection.

Connects to MySQL database on given arguments.

List of parameters for this drivers:

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

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

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

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

Function Parameters:

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

Query-quoted field name.

Query-quoted field name.

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

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

Function Parameters:

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

LIMIT/OFFSET clause for queries.

LIMIT/OFFSET clause for queries.

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

- **Access** public

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

Function Parameters:

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

IOTS_DB method.

IOTS_DB method.

Overwrites PDO method.

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

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

Function Parameters:

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

IOTS_DB method.

IOTS_DB method.

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

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

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

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

Function Parameters:

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

Query-quoted table name.

Query-quoted table name.

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

Class OTS_DB_ODBC

[*line 22*]

ODBC connection interface.

ODBC connection interface.

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

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

Function Parameters:

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

Creates database connection.

Creates database connection.

Connects to ODBC data source on given arguments.

List of parameters for this drivers:

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

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

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

Function Parameters:

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

Query-quoted field name.

Query-quoted field name.

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

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

Function Parameters:

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

LIMIT/OFFSET clause for queries.

LIMIT/OFFSET clause for queries.

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

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

Function Parameters:

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

IOTS_DB method.

IOTS_DB method.

Overwrites PDO method.

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

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

Function Parameters:

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

IOTS_DB method.

IOTS_DB method.

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

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

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

Function Parameters:

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

Query-quoted table name.

Query-quoted table name.

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

Class OTS_DB_PostgreSQL

[line 22]

PostgreSQL connection interface.

PostgreSQL connection interface.

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

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

Function Parameters:

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

Creates database connection.

Creates database connection.

Connects to PgSQL database on given arguments.

List of parameters for this drivers:

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

- *database* - database name.
- *user* - user login.
- *password* - user password.

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

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

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

Query-quoted field name.
 Query-quoted field name.

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

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

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

LIMIT/OFFSET clause for queries.

LIMIT/OFFSET clause for queries.

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

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

Function Parameters:

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

IOTS_DB method.

IOTS_DB method.

Overwrites PDO method.

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

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

Function Parameters:

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

IOTS_DB method.

IOTS_DB method.

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

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

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

Function Parameters:

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

Query-quoted table name.

Query-quoted table name.

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

Class OTS_DB_SQLite

[line 22]

SQLite connection interface.

SQLite connection interface.

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

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

Function Parameters:

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

Creates database connection.

Creates database connection.

Connects to SQLite database on given arguments.

List of parameters for this drivers:

- *database* - database name.

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

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

Function Parameters:

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

Query-quoted field name.

Query-quoted field name.

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

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

Function Parameters:

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

LIMIT/OFFSET clause for queries.

LIMIT/OFFSET clause for queries.

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

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

Function Parameters:

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

IOTS_DB method.

IOTS_DB method.
Overwrites PDO method.

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

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

Function Parameters:

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

IOTS_DB method.

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

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

string function OTS_DB_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 22]

Universal OTServ binary formats reader.

Universal OTServ binary formats reader.

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

OTS_FileLoader::ESCAPE_CHAR

= 0xFD *[line 35]*

Escape another special byte.

Escape another special byte.

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

OTS_FileLoader::NODE_END

= 0xFF *[line 31]*

End of node.
End of node.

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

OTS_FileLoader::NODE_START

= 0xFE *[line 27]*

Start of node.
Start of node.

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

OTS_FileLoader::\$cache

OTS_FileCache = [line 56]

Cache handler.

Cache handler.

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

OTS_FileLoader::\$root

OTS_FileNode = [line 49]

Root node.

Root node.

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

void function OTS_FileLoader::loadFile(\$file) [line 127]

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 116]

Function Parameters:

- [*LOTS 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 82]

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_state(\$properties) [line 98]

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 68]

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 22]

OTServ binary file node representation.

OTServ binary file node representation.

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

string function OTS_FileNode::getBuffer() [line 102]

Returs properties stream.

Returs properties stream.

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

int function OTS_FileNode::getChar() [line 207]

Returns single byte.

Returns single byte.

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

- **Access** public

OTS_FileNode function *OTS_FileNode::getChild()* [*line 143*]

Returns first child.

Returns first child.

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

int function *OTS_FileNode::getLong()* [*line 237*]

Returns quater byte.

Returns quater byte.

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

OTS_FileNode function *OTS_FileNode::getNext()* [*line 123*]

Returns next sibling.

Returns next sibling.

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

int function OTS_FileNode::getShort() [*line 222*]

Returns double byte.

Returns double byte.

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

string function OTS_FileNode::getString([*\$length* = false]) [*line 255*]

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 163*]

Returns node type.

Returns node type.

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

bool function OTS_FileNode::isValid() [*line 183*]

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 112*]

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 153*]

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 133]

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 173]

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 277]

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 62]

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

void function OTS_FileNode::__set_state(\$properties) [line 84]

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 22]

OTServ user group abstraction.

OTServ user group abstraction.

- **Package** POT
- **Version** 0.0.5
- **Version** 0.0.1

- **Since** 0.0.1

int function OTS_Group::count() [*line 385*]

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 348*]

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 154*]

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 241]

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 127]

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 83*]

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 372*]

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 181*]

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 208*]

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 100*]

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 291*]

List of characters in given group.

List of characters in given group.

- **Version** 0.0.5
- **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 321*]

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

bool function OTS_Group::isLoaded() [*line 48*]

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 37*]

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 58]

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 169]

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 267]

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 142*]

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 196]

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 223]

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 115*]

Function Parameters:

- *string* **\$name** Name.

Sets group's name.

Sets group's name.

- **Version** 0.0.1
- **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 22]*

OTServ guild abstraction.

OTServ guild abstraction.

- **Package** POT
- **Version** 0.0.8
- **Version** 0.0.4
- **Since** 0.0.4

void function OTS_Guild::acceptInvite(\$player) [line 442]

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 534]

Function Parameters:

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

Accepts player.

Accepts player.

- **Version** 0.0.4
- **Throws** E_OTTS_NotLoaded If guild is not loaded.
- **Throws** E_OTTS_NoDriver If there is no requests driver assigned.
- **Since** 0.0.4
- **Access** public

int function OTS_Guild::count() [*line 594*]

Returns number of ranks within.

Returns number of ranks within.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E_OTTS_NotLoaded If guild is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::delete() [*line 557*]

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 419]

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 511]

Function Parameters:

- [*OTS Player*](#) **\$player** Player to be rejected.

Deletes request from player.

Deletes request from player.

- **Version** 0.0.4
- **Throws** E_OTTS_NotLoaded If guild is not loaded.
- **Throws** E_OTTS_NoDriver If there is no requests driver assigned.
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::find(\$name) [*line 114*]

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 235*]

Guild creation data.

Guild creation data.

- **Version** 0.0.4
- **Throws** E_OTTS_NotLoaded If guild is not loaded.
- **Since** 0.0.4
- **Access** public

string function OTS_Guild::getCustomField(\$field) [*line 267*]

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 316*]

Reads all ranks that are in this guild.

Reads all ranks that are in this guild.

- **Version** 0.0.5
- **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 346*]

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** 0.0.5
- **Version** 0.0.4
- **Throws** `E_OTS_NotLoaded` If guild is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

int function `OTS_Guild::getId()` [*line 165*]

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 581*]

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 181*]

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 207*]

Returns owning player of this player.

Returns owning player of this player.

- **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 396*]

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 131*]

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 373*]

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 465*]

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 102*]

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 488]

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 141]

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 250]

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 292]

Function Parameters:

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

Writes custom field.

Writes custom field.

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

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

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

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If guild is not loaded.
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::setInvitesDriver([\$invites = null]) [*line 81*]
Function Parameters:

- [*IOTS_GuildAction*](#) **\$invites** Invites driver (don't pass it to clear driver).

Assigns invites handler.
Assigns invites handler.

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

void function OTS_Guild::setName(\$name) [*line 196*]
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 224]

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 91]

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 65*]

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

array function OTS_Guild::__sleep() [*line 53*]

Magic PHP5 method.

Magic PHP5 method.

Allows object serialisation.

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

Class OTS_GuildRank

[*line 22*]

OTServ guild rank abstraction.

OTServ guild rank abstraction.

- **Package** POT
- **Version** 0.0.8
- **Version** 0.0.4
- **Since** 0.0.4

int function OTS_GuildRank::count() [*line 356*]

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 319*]

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 52*]

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 213*]

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_OTTS_NotLoaded If rank is not loaded.
- **Since** 0.0.4
- **Access** public

OTS_Guild function OTS_GuildRank::getGuild() [*line 153*]

Returns guild of this rank.

Returns guild of this rank.

- **Version** 0.0.4
- **Throws** E_OTTS_NotLoaded If rank is not loaded.
- **Since** 0.0.4
- **Access** public

int function OTS_GuildRank::getId() [*line 111*]

Rank ID.

Rank ID.

- **Version** 0.0.4
- **Throws** E_OTTS_NotLoaded If rank is not loaded.
- **Since** 0.0.4
- **Access** public

Iterator function OTS_GuildRank::getIterator() [*line 343*]

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 181*]

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 127*]

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 262*]

Reads all players who has this rank set.

Reads all players who has this rank set.

- **Version** 0.0.5
- **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 292*]

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** 0.0.5
- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If rank is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

bool function OTS_GuildRank::isLoaded() [*line 77*]

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 37*]

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 87*]

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 238]

Function Parameters:

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

Writes custom field.

Writes custom field.

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

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

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

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If rank is not loaded.
- **Since** 0.0.4
- **Access** public

void function OTS_GuildRank::setGuild(\$guild) [line 170]

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 196]

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 142]

Function Parameters:

- *string* **\$name** Name.

Sets rank's name.

Sets rank's name.

- **Version** 0.0.4
- **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_InfoRespond

[line 22]

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

string function OTS_InfoRespond::getClientVersion() [line 121]

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 141]

Returns owner e-mail.

Returns owner e-mail.

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

- **Access** public

string function OTS_InfoRespond::getIP() [*line 49*]

Returns server IP.

Returns server IP.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getLocation() [*line 79*]

Returns server location.

Returns server location.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getMapAuthor() [*line 202*]

Returns map author.

Returns map author.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

int function OTS_InfoRespond::getMapHeight() [*line 222*]

Returns map height.

Returns map height.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getMapName() [*line 191*]

Returns map name.

Returns map name.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

int function OTS_InfoRespond::getMapWidth() [*line 212*]

Returns map width.

Returns map width.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

int function OTS_InfoRespond::getMaxPlayers() [*line 161*]

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 181*]

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 232*]

Returns server's Message Of The Day

Returns server's Message Of The Day

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getName() [*line 59*]

Returns server name.

Returns server name.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

int function OTS_InfoRespond::getOnlinePlayers() [*line 151*]

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 131*]

Returns owner name.

Returns owner name.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

int function OTS_InfoRespond::getPlayersPeak() [*line 171*]

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 69*]

Returns server port.

Returns server port.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getServer() [*line 101*]

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 111*]

Returns server version.

Returns server version.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getTSPQVersion() [*line 29*]

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 39*]

Returns server uptime.

Returns server uptime.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getURL() [*line 89*]

Returns server website.

Returns server website.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

Class OTS_Item

[line 21]

Single item representation.
Single item representation.

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

Constructor *void* function OTS_Item::__construct(\$id) *[line 49]*

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 109*]

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 89*]

Returns item custom attributes.

Returns item custom attributes.

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

int function OTS_Item::getCount() [*line 69*]

Returns count of item.

Returns count of item.

- **Version** 0.0.3

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

int function OTS_Item::getId() [*line 59*]

Returns item type.

Returns item type.

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

void function OTS_Item::setAttributes(\$attributes) [*line 99*]

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 79*]

Function Parameters:

- *int* **\$count** Count.

Sets count of item.

Sets count of item.

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

Class OTS_ItemsList

[line 22]

Items list loader.

Items list loader.

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

OTS_ItemsList::CLIENT_VERSION_750

= 1 *[line 32]*

Tibia client 7.

Tibia client 7.5 version.

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

OTS_ItemsList::CLIENT_VERSION_755

= 2 [*line 36*]

Tibia client 7.

Tibia client 7.55 version.

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

OTS_ItemsList::CLIENT_VERSION_760

= 3 [*line 40*]

Tibia client 7.

Tibia client 7.6 version.

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

OTS_ItemsList::CLIENT_VERSION_770

= 3 [*line 44*]

Tibia client 7.

Tibia client 7.7 version.

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

OTS_ItemsList::CLIENT_VERSION_780

= 4 [*line 48*]

Tibia client 7.

Tibia client 7.8 version.

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

OTS_ItemsList::CLIENT_VERSION_790

= 5 [*line 52*]

Tibia client 7.

Tibia client 7.9 version.

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

OTS_ItemsList::CLIENT_VERSION_792

= 6 [*line 56*]

Tibia client 7.

Tibia client 7.92 version.

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

OTS_ItemsList::CLIENT_VERSION_800

= 7 [*line 60*]

Tibia client 8.

Tibia client 8.0 version.

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

OTS_ItemsList::ITEM_ATTR_CLIENTID

= 17 [*line 69*]

Client ID.

Client ID.

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

OTS_ItemsList::ITEM_ATTR_LIGHT2

= 42 [*line 77*]

Light.

Light.

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

OTS_ItemsList::ITEM_ATTR_SERVERID

= 16 [*line 65*]

Server ID.
Server ID.

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

OTS_ItemsList::ITEM_ATTR_SPEED

= 20 [*line 73*]

Speed.
Speed.

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

OTS_ItemsList::ITEM_ATTR_TOPORDER

= 43 [*line 81*]

Always-on-top order.
Always-on-top order.

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

OTS_ItemsList::ROOT_ATTR_VERSION

= 1 [*line 27*]

Root file attribute.

Root file attribute.

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

int function OTS_ItemsList::count() [*line 476*]

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 486*]

Returns item at current position in iterator.

Returns item at current position in iterator.

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

int function OTS_ItemsList::getBuildVersion() [*line 413*]

Returns build version.

Returns build version.

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

int function OTS_ItemsList::getClientVersion() [*line 403*]

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 424*]

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 446*]

Function Parameters:

- *string* **\$name** Item type name.

Finds item type by it's name.

Finds item type by it's name.

Note: If there are more then one items with same name this function will return first found server ID. It doesn't also mean that it will be the lowest ID - item types are ordered in order that they were loaded from items.xml file.

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

array function OTS_ItemsList::getItemTypesList() [*line 466*]

Returns all loaded items.

Returns all loaded items.

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

int function OTS_ItemsList::getOTBVersion() [*line 393*]

Returns OTB file version.

Returns OTB file version.

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

int function OTS_ItemsList::key() [*line 504*]

Returns ID of current position.

Returns ID of current position.

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

void function OTS_ItemsList::loadItems(\$path) [*line 157*]

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 494]

Moves to next iterator item.

Moves to next iterator item.

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

void function OTS_ItemsList::rewind() [line 522]

Resets iterator index.

Resets iterator index.

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

bool function OTS_ItemsList::valid() [line 514]

Checks if there is anything more in interator.

Checks if there is anything more in iterator.

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

void function OTS_ItemsList::__set_state(\$properties) [line 139]

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 125]

Magic PHP5 method.

Magic PHP5 method.

Allows object unserialisation.

- **Version** 0.0.8

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

Class OTS_ItemType

[line 22]

Item type info.
Item type info.

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

OTS_ItemType::FLAG_ALLOWDISTREAD

= 1048576 *[line 193]*

Can be read from distance.
Can be read from distance.

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

OTS_ItemType::FLAG_ALWAYSONTOP

= 8192 *[line 165]*

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 121*]

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 117*]

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

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

OTS_ItemType::FLAG_BLOCK_SOLID

= 1 [*line 113*]

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

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

OTS_ItemType::FLAG_CANNOTDECAY

= 524288 [*line 189*]

Doesn't decay.
Doesn't decay.

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

OTS_ItemType::FLAG_FLOORCHANGEDOWN

= 256 [*line 145*]

Changes floor under it.
Changes floor under it.

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

OTS_ItemType::FLAG_FLOORCHANGEEST

= 1024 [*line 153*]

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 149*]

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 157*]

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 161*]

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 177*]

Can be hang(?).

Can be hang(?).

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

OTS_ItemType::FLAG_HAS_HEIGHT

= 8 [*line 125*]

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 185*]

Is oriented horizontally.
Is oriented horizontally.

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

OTS_ItemType::FLAG_MOVEABLE

= 64 [*line 137*]

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

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

OTS_ItemType::FLAG_PICKUPABLE

= 32 [*line 133*]

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 169*]

Has readable sign.

Has readable sign.

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

OTS_ItemType::FLAG_ROTABLE

= 32768 [*line 173*]

Can be rotated by player.

Can be rotated by player.

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

OTS_ItemType::FLAG_STACKABLE

= 128 [*line 141*]

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 129*]

Can be used by players.

Can be used by players.

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

OTS_ItemType::FLAG_VERTICAL

= 131072 [*line 181*]

Is oriented vertically.

Is oriented vertically.

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

OTS_ItemType::ITEM_GROUP_AMMUNITION

= 4 [*line 43*]

Ammunition.

Ammunition.

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

OTS_ItemType::ITEM_GROUP_ARMOR

= 5 *[line 47]*

Armor.

Armor.

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

OTS_ItemType::ITEM_GROUP_CONTAINER

= 2 *[line 35]*

Container.

Container.

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

OTS_ItemType::ITEM_GROUP_DOOR

= 13 *[line 79]*

Door.

Door.

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

OTS_ItemType::ITEM_GROUP_FLUID

= 12 [*line 75*]

Liquid thing.

Liquid thing.

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

OTS_ItemType::ITEM_GROUP_GROUND

= 1 [*line 31*]

Ground tile.

Ground tile.

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

OTS_ItemType::ITEM_GROUP_KEY

= 10 [*line 67*]

Key.

Key.

- **Version** 0.0.8

- **Since** 0.0.8

OTS_ItemType::ITEM_GROUP_MAGICFIELD

= 8 *[line 59]*

Magic field.

Magic field.

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

OTS_ItemType::ITEM_GROUP_NONE

= 0 *[line 27]*

No group speciffied.

No group speciffied.

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

OTS_ItemType::ITEM_GROUP_RUNE

= 6 *[line 51]*

Rune.

Rune.

- **Version** 0.0.8

- **Since** 0.0.8

OTS_ItemType::ITEM_GROUP_SPLASH

= 11 [*line 71*]

Splash effect.
 Splash effect.

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

OTS_ItemType::ITEM_GROUP_TELEPORT

= 7 [*line 55*]

Teleport field.
 Teleport field.

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

OTS_ItemType::ITEM_GROUP_WEAPON

= 3 [*line 39*]

Weapon.
 Weapon.

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

OTS_ItemType::ITEM_GROUP_WRITEABLE

= 9 [*line 63*]

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 100*]

Container.

Container.

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

OTS_ItemType::ITEM_TYPE_DEPOT

= 1 [*line 88*]

Depot locker.

Depot locker.

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

OTS_ItemType::ITEM_TYPE_DOOR

= 5 [*line 104*]

Door.

Door.

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

OTS_ItemType::ITEM_TYPE_MAGICFIELD

= 6 [*line 108*]

Magic field.

Magic field.

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

OTS_ItemType::ITEM_TYPE_MAILBOX

= 2 [*line 92*]

Mailbox.

Mailbox.

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

OTS_ItemType::ITEM_TYPE_NONE

= 0 [*line 84*]

No special type.
No special type.

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

OTS_ItemType::ITEM_TYPE_TRASHHOLDER

= 3 [*line 96*]

Trash can.
Trash can.

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

Constructor *void* function OTS_ItemType::__construct(\$id) [*line 249*]

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 539*]

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 331*]

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 359*]

Returns all attributes list.

Returns all attributes list.

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

int function OTS_ItemType::getClientId() [*line 290*]

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 369*]

Returns group.

Returns group.

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

int function OTS_ItemType::getId() [*line 280*]

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 310*]

Returns item name.

Returns item name.

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

int function OTS_ItemType::getType() [*line 389*]

Returns item type.

Returns item type.

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

bool function OTS_ItemType::hasHeight() [*line 429*]

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 479*]

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 419*]

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 509*]

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 529*]

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 459*]

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 449*]

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 489*]

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 499*]

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 469*]

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 439*]

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 519*]

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 349*]

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 300*]

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 409*]

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 379]

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 320]

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 399]

Function Parameters:

- *int* **\$type** Type.

Sets item type.

Sets item type.

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

void function OTS_ItemType::__set_state(\$properties) [line 262]

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 22]

Map position point.
Map position point.

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

Constructor *void* function OTS_MapCoords::__construct(\$x, \$y, \$z) *[line 52]*

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 77*]

Returns X.

Returns X.

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

int function OTS_MapCoords::getY() [*line 87*]

Returns Y.

Returns Y.

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

int function OTS_MapCoords::getZ() [*line 97*]

Returns Z.

Returns Z.

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

void function OTS_MapCoords::__set_state(\$properties) [line 67]

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 23]

Wrapper for monsters files DOMDocument.

Wrapper for monsters files 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
- **Version** 0.0.6
- **Since** 0.0.6

int function OTS_Monster::getArmor() [*line 269*]

Returns monster armor.

Returns monster armor.

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

array function OTS_Monster::getAttacks() [*line 310*]

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 251*]

Returns monster defense rate.

Returns monster defense rate.

- **Version** 0.0.6

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

array function OTS_Monster::getDefenses() [*line 287*]

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 50*]

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 119*]

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 98*]

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 88*]

Returns monster HP.
Returns monster HP.

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

array function OTS_Monster::getImmunities() [*line 194*]

Returns all monster immunities.
Returns all monster immunities.

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

array function OTS_Monster::getLoot() [*line 164*]

Returns all possible loot.

Returns all possible loot.

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

int/bool function OTS_Monster::getManaCost() [*line 70*]

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 30*]

Returns monster name.

Returns monster name.

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

string function OTS_Monster::getRace() [*line 40*]

Returns monster race.

Returns monster race.

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

int function OTS_Monster::getSpeed() [*line 60*]

Returns monster speed.

Returns monster speed.

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

array function OTS_Monster::getVoices() [*line 140*]

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 225*]

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

Class OTS_OTBMFile

[*line 26*]

OTBM format reader.

OTBM format reader.

- **Package** POT
- **Version** 0.0.8
- **Version** 0.0.6

- **Since** 0.0.6

OTS_OTBMFile::OTBM_ATTR_ACTION_ID

= 4 *[line 43]*

Action ID.

Action ID.

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

OTS_OTBMFile::OTBM_ATTR_DEPOT_ID

= 10 *[line 67]*

Depot ID.

Depot ID.

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

OTS_OTBMFile::OTBM_ATTR_DESC

= 7 *[line 55]*

Description.

Description.

- **Version** 0.0.6

- **Since** 0.0.6

OTS_OTBMFile::OTBM_ATTR_DESCRIPTION

= 1 [*line 31*]

Description attribute.
Description attribute.

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

OTS_OTBMFile::OTBM_ATTR_EXT_FILE

= 2 [*line 35*]

External file.
External file.

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

OTS_OTBMFile::OTBM_ATTR_EXT_HOUSE_FILE

= 13 [*line 79*]

External houses file.
External houses file.

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

OTS_OTBMFile::OTBM_ATTR_EXT_SPAWN_FILE

= 11 [*line 71*]

External spawns file.

External spawns file.

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

OTS_OTBMFile::OTBM_ATTR_HOUSEDOORID

= 14 [*line 83*]

ID of doors.

ID of doors.

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

OTS_OTBMFile::OTBM_ATTR_ITEM

= 9 [*line 63*]

Item.

Item.

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

OTS_OTBMFile::OTBM_ATTR_RUNE_CHARGES

= 12 [*line 75*]

Rune changes amount.
Rune changes amount.

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

OTS_OTBMFile::OTBM_ATTR_TELE_DEST

= 8 [*line 59*]

Teleport destination.
Teleport destination.

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

OTS_OTBMFile::OTBM_ATTR_TEXT

= 6 [*line 51*]

Text.
Text.

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

OTS_OTBMFile::OTBM_ATTR_TILE_FLAGS

= 3 [*line 39*]

Tile flags.

Tile flags.

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

OTS_OTBMFile::OTBM_ATTR_UNIQUE_ID

= 5 [*line 47*]

Unique ID.

Unique ID.

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

OTS_OTBMFile::OTBM_NODE_HOusetile

= 14 [*line 140*]

Tile of house.

Tile of house.

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

OTS_OTBMFile::OTBM_NODE_ITEM

= 6 [*line 108*]

Item.

Item.

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

OTS_OTBMFile::OTBM_NODE_ITEM_DEF

= 3 [*line 96*]

Item definition.

Item definition.

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

OTS_OTBMFile::OTBM_NODE_MAP_DATA

= 2 [*line 92*]

Map data container.

Map data container.

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

OTS_OTBMFile::OTBM_NODE_MONSTER

= 11 [*line 128*]

Monster.

Monster.

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

OTS_OTBMFile::OTBM_NODE_ROOTV1

= 1 [*line 88*]

Root node.

Root node.

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

OTS_OTBMFile::OTBM_NODE_SPAWNS

= 9 [*line 120*]

Spawns container.

Spawns container.

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

OTS_OTBMFile::OTBM_NODE_SPAWN_AREA

= 10 [*line 124*]

Spawn.
Spawn.

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

OTS_OTBMFile::OTBM_NODE_TILE

= 5 [*line 104*]

Single tile.
Single tile.

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

OTS_OTBMFile::OTBM_NODE_TILE_AREA

= 4 [*line 100*]

Map tiles fragment.
Map tiles fragment.

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

OTS_OTBMFile::OTBM_NODE_TILE_REF

= 8 [*line 116*]

Tile reference.

Tile reference.

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

OTS_OTBMFile::OTBM_NODE_TILE_SQUARE

= 7 [*line 112*]

Tile.

Tile.

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

OTS_OTBMFile::OTBM_NODE_TOWN

= 13 [*line 136*]

Town.

Town.

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

OTS_OTBMFile::OTBM_NODE_TOWNS

= 12 [*line 132*]

Towns container.

Towns container.

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

int function OTS_OTBMFile::count() [*line 420*]

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 432*]

Returns town at current position in iterator.

Returns town at current position in iterator.

- **Version** 0.0.8
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.0.8
- **Access** public

string function OTS_OTBMFile::getDescription() [*line 351*]

Returns map description.

Returns map description.

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

int function OTS_OTBMFile::getHeight() [*line 341*]

Returns map height.

Returns map height.

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

int|bool function OTS_OTBMFile::getTownID(\$name) [*line 362*]

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 373*]

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 390*]

Returns list (id => name) of loaded towns.

Returns list (id => name) of loaded towns.

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

OTS_MapCoords|bool function OTS_OTBMFile::getTownTemple(\$id) [*line 401*]

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 331*]

Returns map width.

Returns map width.

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

int function OTS_OTBMFile::key() [*line 455*]

Returns ID of current position.

Returns ID of current position.

- **Version** 0.0.8
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.0.8
- **Access** public

void function OTS_OTBMFile::loadFile(\$file) [*line 216*]

Function Parameters:

- *string* **\$file** Filename.

Loads OTBM file content.

Loads OTBM file content.

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

void function OTS_OTBMFile::next() [*line 443*]

Moves to next iterator town.

Moves to next iterator town.

- **Version** 0.0.8
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.0.8
- **Access** public

void function OTS_OTBMFile::rewind() [*line 478*]

Resets iterator index.

Resets iterator index.

- **Version** 0.0.8
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.0.8
- **Access** public

bool function OTS_OTBMFile::valid() [*line 467*]

Checks if there is anything more in interator.

Checks if there is anything more in interator.

- **Version** 0.0.8
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.0.8
- **Access** public

void function OTS_OTBMFile::__set_state(\$properties) [line 198]

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 184]

Magic PHP5 method.

Magic PHP5 method.

Allows object unserialisation.

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

Class OTS_Player

[line 23]

OTServ character abstraction.

OTServ character abstraction.

- **Package** POT
- **Version** 0.0.7
- **Version** 0.0.1
- **Since** 0.0.1

void function OTS_Player::ban([\$time = 0]) [line 1673]

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 1727]

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 85*]

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 187*]

Returns account of this player.

Returns account of this player.

- **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::getCap() [*line 842*]

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

mixed function OTS_Player::getConditions() [*line 980*]

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 1279*]

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 1554*]

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

int function OTS_Player::getDirection() [*line 572*]

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 329*]

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 216*]

Returns group of this player.

Returns group of this player.

- **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::getGuildNick() [*line 1067*]

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 410*]

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 437*]

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 143*]

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 896*]

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 869*]

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 356*]

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 734*]

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 599*]

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 626*]

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 653*]

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 680*]

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 707*]

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 1190*]

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 1217*]

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 1244*]

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 383*]

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 464*]

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 491*]

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 518*]

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 160*]

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 761*]

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 788*]

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 815*]

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 246*]

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 1111*]

Assigned guild rank.

Assigned guild rank.

- **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 1095*]

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 1007*]

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 952*]

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 275*]

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 1334*]

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 1366*]

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 1419*]

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

int function OTS_Player::getSoul() [*line 545*]

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 1163*]

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 302*]

Player proffesion.

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 1749*]

Player proffesion name.

Player proffesion name.

- **Version** 0.0.6
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.6
- **Since** 0.0.1
- **Access** public

bool function OTS_Player::hasRedSkull() [*line 1034*]

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 1708*]

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 102*]

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 923*]

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 63]

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 112]

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 204]

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 857]

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 995]

Function Parameters:

- *mixed* **\$conditions** Condition binary field.

Sets conditions.

Sets conditions.

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

`void function OTS_Player::setCustomField($field, $value) [line 1309]`

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 improper queries that could lead to SQL injection will be executed, but it can make your code working wrong way. For example: `$object->setCustomField('foo', '1');` will quote 1 as as string ('1') instead of passing it as a integer.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.3

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

void function OTS_Player::setDepot(\$depot, [\$item = null], [\$pid = 0], [\$depot_id = 0]) [line 1609]

Function Parameters:

- *int* **\$depot** Depot ID to save items.
- [*OTS_Item*](#) **\$item** Item (can be a container with content) for given depot. Leave this parameter blank to clear depot.
- *int* **\$pid** Deprecated, not used anymore.
- *int* **\$depot_id** Internal, for further use.

Sets depot content.

Sets depot content.

- **Version** 0.0.5
- **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 587]

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 344*]

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 233*]

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 1082]

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 425]

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 452]

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 911]

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 884]

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 371*]

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 749*]

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 614]

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 641]

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 668]

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 695]

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 722]

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 1205]

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 1232]

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 1259]

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 398]

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 479*]

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 506*]

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 533]

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 175]

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 776]

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 803]

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 830]

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 263]

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 1144]

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 1134]

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 1055]

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 1022]

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 968]

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 290]

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 1352]

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 1384*]

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 1473*]

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_OTTS_NotLoaded If player is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setSoul(\$soul) [line 560]

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 1178]

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 317]

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 1690]

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 1047*]

Unsets red skull flag.

Unsets red skull flag.

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

void function OTS_Player::unsetSave() [*line 938*]

Unsets save flag.

Unsets save flag.

- **Version** 0.0.7
- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

array function OTS_Player::__sleep() [*line 52*]

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 21*]

Wrapper for spell info.

Wrapper for spell info.

- **Package** POT
- **Version** 0.0.7
- **Since** 0.0.7

Constructor *void* function OTS_Spell::__construct(\$type, \$spell) [*line 43*]

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

int function OTS_Spell::getCharges() [*line 104*]

Number of rune charges.

Number of rune charges.

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

int function OTS_Spell::getConjureCount() [*line 224*]

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 204*]

Returns ID of item conjured by this spell.

Returns ID of item conjured by this spell.

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

int function OTS_Spell::getID() [*line 74*]

Returns rune item id.

Returns rune item id.

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

int function OTS_Spell::getLevel() [*line 114*]

Level required for use.

Level required for use.

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

int function OTS_Spell::getMagicLevel() [*line 124*]

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 134*]

Mana cost.

Mana cost.

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

string function OTS_Spell::getName() [*line 64*]

Returns spell name.

Returns spell name.

- **Version** 0.0.7

- **Since** 0.0.7
- **Access** public

int function OTS_Spell::getReagentId() [*line 214*]

Returns ID of item that is used by spell.

Returns ID of item that is used by spell.

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

int function OTS_Spell::getSoul() [*line 144*]

Soul points cost.

Soul points cost.

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

int function OTS_Spell::getType() [*line 54*]

Returns spell type.

Returns spell type.

- **Version** 0.0.7
- **Since** 0.0.7

- **Access** public

array function OTS_Spell::getVocations() [*line 234*]

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 84*]

Returns spell formula.

Returns spell formula.

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

bool function OTS_Spell::hasParams() [*line 154*]

Checks if spell has parameter.

Checks if spell has parameter.

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

bool function OTS_Spell::isAggresive() [*line 94*]

Checks if spell is threatened as unfriendly by other creatures.

Checks if spell is threatened as unfriendly by other creatures.

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

bool function OTS_Spell::isEnabled() [*line 164*]

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 174*]

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 194*]

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 184*]

Checks if spell requires PACC.

Checks if spell requires PACC.

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

Class OTS_SQLField

[*line 20*]

SQL identifier representation.

SQL identifier representation.

- **Package** POT

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

Constructor *void* function OTS_SQLField::__construct(\$name, [\$table = "]) [*line 41*]

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 52*]

Returns field name.

Returns field name.

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

string function OTS_SQLField::getTable() [*line 62*]

Returns table name.

Returns table name.

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

Class OTS_SQLFilter

[line 20]

SQL WHERE clause object.
SQL WHERE clause object.

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

OTS_SQLFilter::CRITERIUM_AND

= 1 *[line 58]*

AND sibling.
AND sibling.

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

OTS_SQLFilter::CRITERIUM_OR

= 2 [*line 62*]

OR sibling.

OR sibling.

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

OTS_SQLFilter::OPERATOR_EQUAL

= 1 [*line 25*]

Equal operator.

Equal operator.

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

OTS_SQLFilter::OPERATOR_GREATER

= 3 [*line 33*]

Greater-then operator.

Greater-then operator.

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

OTS_SQLFilter::OPERATOR_LIKE

= 7 *[line 49]*

LIKE operator.

LIKE operator.

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

OTS_SQLFilter::OPERATOR_LOWER

= 2 *[line 29]*

Lower-then operator.

Lower-then operator.

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

OTS_SQLFilter::OPERATOR_NEQUAL

= 4 *[line 37]*

Not-equal operator.

Not-equal operator.

- **Version** 0.0.5

- **Since** 0.0.5

OTS_SQLFilter::OPERATOR_NGREATER

= 6 [*line 45*]

Not-greater-then operator.

Not-greater-then operator.

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

OTS_SQLFilter::OPERATOR_NLIKE

= 8 [*line 53*]

Not-LIKE operator.

Not-LIKE operator.

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

OTS_SQLFilter::OPERATOR_NLOWER

= 5 [*line 41*]

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 238*]

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 251*]

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 263]

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

array function OTS_SQLFilter::__sleep() [line 79]

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 92*]

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 POT

[*line 36*]

Main POT class.

Main POT class.

- **Package** POT
- **Version** 0.0.7
- **Version** 0.0.1
- **Since** 0.0.1

POT::BAN_ACCOUNT

= 3 [*line 267*]

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 253*]

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 260*]

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 41]*

MySQL driver.
MySQL driver.

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

POT::DB_ODBC

= 4 *[line 59]*

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 52]*

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 45]*

SQLite driver.

SQLite driver.

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

POT::DEPOT_SID_FIRST

= 100 *[line 245]*

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 108*]

East.

East.

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

POT::DIRECTION_NORTH

= 0 [*line 104*]

North.

North.

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

POT::DIRECTION_SOUTH

= 2 [*line 112*]

South.

South.

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

POT::DIRECTION_WEST

= 3 [*line 116*]

West.

West.

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

POT::ORDER_ASC

= 1 [*line 275*]

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 282*]

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 64*]

Female gender.

Female gender.

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

POT::SEX_MALE

= 1 [*line 68*]

Male gender.

Male gender.

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

POT::SKILL_AXE

= 3 [*line 145*]

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 131*]

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 152*]

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 166*]

Fishing.
Fishing.

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

POT::SKILL_FIST

= 0 [*line 124*]

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 159*]

Shielding.

Shielding.

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

POT::SKILL_SWORD

= 2 [*line 138*]

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 237*]

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 195*]

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 188*]

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 223]*

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 174]*

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 209]*

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 216*]

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 181*]

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 202*]

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 230*]

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 304*]

Conjure spell.

Conjure spell.

- **Version** 0.0.7
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.7

POT::SPELL_INSTANT

= 1 [*line 297*]

Instant spell.

Instant spell.

- **Version** 0.0.7
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.7

POT::SPELL_RUNE

= 0 [*line 290*]

Rune spell.

Rune spell.

- **Version** 0.0.7
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.7

POT::VOCATION_DRUID

= 2 *[line 87]*

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 99]*

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 75]*

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 93*]

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 81*]

Sorcerer.

Sorcerer.

- **Version** 0.0.1
- **Deprecated** 0.0.5 Vocations are now loaded dynamicly from vocations.xml file.
- **Since** 0.0.1

void function POT::banIP(\$ip, [\$mask = '255.255.255.255'], [\$time = 0]) [*line 645*]

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 427*]

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 737*]

Creates lists filter.

Creates lists filter.

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

IOTS_DAO function POT::createObject(\$class) [*line 478*]

Function Parameters:

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

Creates OTServ DAO class instance.

Creates OTServ DAO class instance.

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

OTS_Spell function POT::getConjure(\$name) [*line 967*]

Function Parameters:

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

Returns given conjure spell.

Returns given conjure spell.

- **Version** 0.0.7

- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.7
- **Access** public

array function POT::getConjuresList() [*line 954*]

Returns list of conjure spells.

Returns list of conjure spells.

- **Version** 0.0.7
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.7
- **Access** public

PDO function POT::getDBHandle() [*line 553*]

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

POT function POT::getInstance() [*line 311*]

Singleton.

Singleton.

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

OTS_Spell|null function POT::getInstant(\$name) [*line 935*]

Function Parameters:

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

Returns given instant spell.

Returns given instant spell.

- **Version** 0.0.7
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.7
- **Access** public

array function POT::getInstantsList() [*line 922*]

Returns list of instants.

Returns list of instants.

- **Version** 0.0.7
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.7
- **Access** public

OTS_Monster|null function POT::getMonster(\$name) [*line 808*]

Function Parameters:

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

Returns loaded data of given monster.

Returns loaded data of given monster.

- **Version** 0.0.7
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.6
- **Access** public

array function POT::getMonstersList() [*line 795*]

Returns list of laoded monsters.

Returns list of laoded monsters.

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

OTS_Spell *null* function POT::getRune(\$name) [*line 903*]

Function Parameters:

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

Returns given rune spell.

Returns given rune spell.

- **Version** 0.0.7
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.7
- **Access** public

array function POT::getRunesList() [*line 890*]

Returns list of runes.

Returns list of runes.

- **Version** 0.0.7
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.7
- **Access** public

int|bool function POT::getVocationID(\$name) [*line 597*]

Function Parameters:

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

Returns vocation's ID.

Returns vocation's ID.

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

string|bool function POT::getVocationName(\$id) [*line 610*]

Function Parameters:

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

Returns name of given vocation's ID.

Returns name of given vocation's ID.

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

array function POT::getVocationsList() [*line 629*]

Returns list (id => name) of loaded vocations.

Returns list (id => name) of loaded vocations.

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

bool function POT::isIPBanned(\$ip) [*line 713*]

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

void function POT::loadClass(\$class) [line 386]

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::loadMonsters(\$path) [line 767]

Function Parameters:

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

Loads monsters mapping file.

Loads monsters mapping file.

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

void function POT::loadSpells(\$file) [line 858]

Function Parameters:

- *string* **\$file** Spells file name.

Loads spells list.

Loads spells list.

- **Version** 0.0.7
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.7
- **Access** public

void function POT::loadVocations(\$file) [line 576]

Function Parameters:

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

Loads vocations list.

Loads vocations list.

Loads vocations list from given file.

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

OTS_InfoRespond|bool function POT::serverStatus(\$server, \$port) [line 496]

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::setPOTPath(\$path) [*line 342*]

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 680]

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

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** 0.0.2
- **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_OTBMErrors](#)

E_OTS_NoDriver

- Exception
 - [E_OTS_NoDriver](#)

E_OTS_NotLoaded

- Exception
 - [E_OTS_NotLoaded](#)

E_OTS_OutOfBuffer

- Exception
 - [E_OTS_OutOfBuffer](#)

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_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_InfoRespond

- DOMDocument
 - [OTS_InfoRespond](#)

OTS_Item

- [OTS_Item](#)
 - [OTS_Container](#)

OTS_ItemType

- [OTS_ItemType](#)

OTS_MapCoords

- [OTS_MapCoords](#)

OTS_Monster

- DOMDocument
 - [OTS_Monster](#)

OTS_Spell

- [OTS_Spell](#)

OTS_SQLField

- [OTS_SQLField](#)

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

[0.0.8]

- * Added items.xml and items.otb files support. <wrzasq>
- * Added Iterator and Countable interfaces implementation in OTS_OTBMFile class. <wrzasq>
- * Fixed quoting guild rank name while updating. <wrzasq>
- * Fixed custom fields reading in OTS_Guild class. <wrzasq>

[0.0.7]

- * Updated for last database changes. <wrzasq>
- * Even more PHP 5.0 compatibility. <wrzasq>
- * Added spells support. <wrzasq>
- * Added possibility to sort lists by custom fields, not only selected. <wrzasq>
- * Fixed typos. <wrzasq>

[0.0.6]

- * Updated for last database changes. <wrzasq>
- * Increased PHP 5.0 compatibility. <wrzasq>
- * Added generic binary formats reader with cache drivers. <wrzasq>
- * Added OTBM files basic support. <wrzasq>
- * Added monsters support. <wrzasq>
- * Added OTS_Player::getVocationName() method. <wrzasq>

[0.0.5]

- * Added support for vocations.xml file. <wrzasq>
- * Added support for bans. <wrzasq>
- * Added sorting and filtering for lists. <wrzasq>
- * Code grouped into base classes. <wrzasq>
- * Some code optimisation. <wrzasq>
- * Fixed typos. <wrzasq>

[0.0.4]

- * Added guild system support (guilds, ranks, invitations and requests drivers mechanisms). <wrzasq>
- * Added account group support. <wrzasq>
- * Added support for depot_id field (it is reserved in OTServ for further use). <wrzasq>
- * Added PostgreSQL and ODBC drivers. <wrzasq>
- * Added __sleep() and __wakeup() methods to allow POT objects to be stored in sessions. <wrzasq>
- * Added __clone() methods to allow save ID-losing cloning of POT objects. <wrzasq>
- * Added __set_state() methods. <wrzasq>
- * Updated players table structure. <wrzasq>
- * Dropped REGEXP operator bindings - not used anywhere. <wrzasq>
- * Fixed items loading and saving. <wrzasq>
- * Fixed typos. <wrzasq>

[0.0.3]

- * Added custom fields support. <wrzasq>
- * Added items and depots support. <wrzasq>
- * Added support for players PACC timestamps. <wrzasq>
- * Fixed loading skills. <wrzasq>
- * Replaced E_USER_* with exceptions. <wrzasq>
- * Uses fetchAll() in loops to prevent MySQL buffering problems. <wrzasq>
- * Restricted access to POT class constructor to make sure it won't be instantiated directly. <wrzasq>

[0.0.2]

- * Added "compat" library for POT. <wrzasq>
- * Added skills support in OTS_Player class. <wrzasq>
- * Added 'info' serverStatus() method and respond handler for server status protocol. <wrzasq>

- * Fixed `redskulltime` field name in OTS_Player. <wrzasq>
- * Fixed 'password' parameter for DB_MYSQL driver. <wrzasq>
- * Added find() to OTS_Account class to load accounts by their's e-mail addresses. <wrzasq>
- * POT class now 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 langauge structure and has it's limitations!).

[4] Don't use functions - use class methods (except Compat package).

This will allow __autoload() handling for all routines as they will be members of classes.

[5] Use 4 spaces as tabulation.

Tabulator character can be differently displayed and generally four spaces makes code more readable.

[6] Always use brackets for blocks and leave them in separated lines in same nesting level that block instruction:

```
if(condition)
{
    for($i = 0; $i < $j; $i++)
    {
        statement;
    }
}
```

[7] Use single quotes inseed of double ones.

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

[8] Use spaces between parenthess and operators (except object member accessing operator):

```
$foo = $lol . $rotfl;  
$foo .= $bar;  
$obj = new Class( substr( str_replace( implode('.', $array), ',', '.'), 2) );  
echo $obj->field;  
echo $obj->method( rand() );  
echo $obj->method($value);
```

[9] Use `<?php` opening tag.

It is the most reliable and standard way for starting PHP code.

[10] Use `isset(array[offset])` instead of `array_key_exists()`.

It saves a lot of resources (relatively).

II. Versioning:

- * Major - Major toolkit milestone.
- * Minor - Toolkit noticeable step.
- * Release - Independent package version.

Package version shouldn't be increased if package itself wasn't changed - but it can't increase its major/minor numbers over current toolkit release.

III. 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://www.otserv-aac.info/pot/> 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 0.0.6 version?

- * Updated for last database changes.

There were minor database changes like save counter which this POT version of course supports.

- * Increased PHP 5.0 compatibility.

Dropper array type hints. POT should now run on PHP 5.0 easily.

- * Added generic binary formats reader with cache drivers.

Universal class for reading OTServ binary file formats (OTB-based).

- * Added OTBM files basic support.

Support for OTBM map files. Currently it saves only spawns points, but as it contains entire map info you can easily add map tiles reading for example. It also supports cache mechanism, so you it won't slow down your scripts if you have big OTBM files.

- * Added monsters support.

You can now load list of monsters and easily get info about particular monsters using OTS_Monster wrapper for DOMDocument.

- * Added OTS_Player::getVocationName() method.

Wrapper for POT::getVocationName() with current player vocation ID.

Appendix D - Todo List

In Package POT

In [OTS.php](#)

- 0.0.9: Houses support.
- 0.1.0: Get rid of POT::getInstance()->create*() calls - use POT::getInstance()->getDBHandle() in constructors.
- 0.1.0: Handle instances of loaded data/ directory resources to allow exchange data between classes and objects.
- 0.1.0: Implement __get()/__set()/__call()/__toString(); ArrayAccess interface.
- 0.1.0: Provide suite for automatic items, spells, monsters and other data/ directory info loading.
- 0.1.0: Put items map handling into POT class or separate vocations, monsters and spells too.
- 1.0.0: Complete phpUnit test.
- 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: OTAdmin protocol.
- 1.0.0: PHAR and PHK packages.
- 1.0.0: SOAP interface for remote controll.

In [OTS Item.php](#)

- 0.1.0: Provide item type fetching.

In [OTS Monster.php](#)

- 0.1.0: Fetching loot as item types.

In [OTS_OTBMFile.php](#)

- 0.1.0: Houses support.
- 1.0.0: Complete OTBM support: link tiles with items, spawns and houses.
- 1.0.0: Spawns support.

In [OTS_Player.php](#)

- 0.1.0: Check item types if they are containers during loading slots/depots.

In [OTS_Spell.php](#)

- 0.1.0: Link conjures with item types when items support will be added.

Index

A

About OTServ AAC scripts	21
Account number hack	18

C

constructor OTS_MapCoords::__construct()	221
<i>Sets coords for point.</i>	
constructor OTS_ItemType::__construct()	210
<i>Initializes new item type object.</i>	
constructor OTS_Item::__construct()	182
<i>Creates item of given ID.</i>	
constructor OTS_Spell::__construct()	293
<i>Sets spell info.</i>	
constructor OTS_SQLField::__construct()	300
<i>Creates new field representation.</i>	
CHANGELOG	348
compat.php	338
<i>POT compatibility assurance package.</i>	
constructor OTS_DB_SQLite::__construct()	124
<i>Creates database connection.</i>	
constructor OTS_DB_PostgreSQL::__construct()	120
<i>Creates database connection.</i>	
constructor IOTS_GuildAction::__construct()	80
<i>Objects are initialized with a guild that they are assigned to.</i>	
constructor IOTS_DB::__construct()	74
<i>Connection parameters.</i>	
constructor IOTS_DAO::__construct()	73
<i>DAO objects must be initialized with a database.</i>	
constructor OTS_Base_DAO::__construct()	100
<i>Sets database connection handler.</i>	
constructor OTS_Base_List::__construct()	103
<i>Sets database connection handler.</i>	
constructor OTS_DB_ODBC::__construct()	117
<i>Creates database connection.</i>	
constructor OTS_DB_MySQL::__construct()	113
<i>Creates database connection.</i>	
constructor E_OTSErrorCode::__construct()	67
<i>Sets error code.</i>	

D

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

E

E_OTS_NoDriver	71
<i>Occurs when code attempts to execute driven action that has no assigned driver to handle it.</i>	
E_OTS_FileLoaderError::ERROR_TELL_ERROR	70
<i>Failed to read position in file.</i>	
E_OTS_FileLoaderError::ERROR_SEEK_ERROR	70
<i>Failed to seek in given position in file.</i>	
E_OTS_FileLoaderError::ERROR_NOT_OPEN	69
<i>Attempted to execute operation on not opened file.</i>	
E_OTS_NotLoaded	71
<i>Occurs when code attempts to access property of not loaded object.</i>	
E_OTS_OTBMErrors	72
<i>OTBM map loading error.</i>	
E_OTS_OutOfBuffer	73
<i>Occurs when properties stream has ended and there is still read attempt.</i>	
E_OTS_OTBMErrors::LOADMAPERROR_UNKNOWNNODETYPE	72
<i>Unknown node type.</i>	
E_OTS_OTBMErrors::LOADMAPERROR_OUTDATEDHEADER	72
<i>Unsupported file version.</i>	
E_OTS_FileLoaderError::ERROR_INVALID_FORMAT	69
<i>File corrupted.</i>	
E_OTS_FileLoaderError::ERROR_INVALID_FILE_VERSION	69
<i>Unsupported file version.</i>	
E_OTS_OTBMErrors.php	29
<i>Code in this file bases on original OTServ OTBM format loading C++ code (iomapotbm).</i>	
E_OTS_NotLoaded.php	28
E_OTS_NoDriver.php	27
E_OTS_FileLoaderError.php	26
<i>Code in this file bases on original OTServ binary format loading C++ code (fileloader).</i>	
E_OTS_OutOfBuffer.php	30
<i>Code in this file bases on original OTServ binary format loading C++ code (fileloader).</i>	
E_OTS_ErrorCode	67
<i>Generic exception class for error codes.</i>	
E_OTS_FileLoaderError::ERROR_EOF	68
<i>Unexpected end of file.</i>	
E_OTS_FileLoaderError::ERROR_CAN_NOT_OPEN	68
<i>Could not open file.</i>	
E_OTS_FileLoaderError	68
<i>Error during reading OTServ binary file.</i>	
E_OTS_ErrorCode.php	25

G

Guild action drivers	15
Guilds	13

I

IOTS_GuildAction::addRequest()	80
<i>Adds new request.</i>	

IOTS_GuildAction::deleteRequest()	81
<i>Deletes request.</i>	
IOTS_GuildAction	79
<i>Guild action interface.</i>	
IOTS_FileCache::writeCache()	78
<i>Writes node cache.</i>	
IOTS_FileCache::readCache()	78
<i>Returns cache.</i>	
IOTS_GuildAction::listRequests()	81
<i>List of saved pending actions.</i>	
IOTS_GuildAction::submitRequest()	81
<i>Finalizes request.</i>	
INSTALL	352
IOTS_ItemsCache::writeItems()	83
<i>Writes items cache.</i>	
IOTS_ItemsCache::readItems()	82
<i>Returns cache.</i>	
IOTS_ItemsCache	82
<i>This interface defines items.</i>	
IOTS_FileCache	78
<i>This interface describes binary files cache control drivers.</i>	
IOTS_DB::tableName()	77
<i>Query-quoted table name.</i>	
IOTS_ItemsCache.php	35
IOTS_DAO	73
<i>OTServ database object.</i>	
IOTS_GuildAction.php	34
IOTS_FileCache.php	33
IOTS_DB.php	32
IOTS_DB	74
<i>OTServ database handler interface.</i>	
IOTS_DB::fieldName()	75
<i>Query-quoted field name.</i>	
IOTS_DB::SQLquote()	77
<i>Query-quoted string value.</i>	
IOTS_DB::SQLquery()	76
<i>Evaluates query.</i>	
IOTS_DB::limit()	76
<i>LIMIT/OFFSET clause for queries.</i>	
IOTS_DB::lastInsertId()	75
<i>ID of last created record.</i>	
IOTS_DAO.php	31

L

LICENSE	346
-------------------------	-----

N

NEWS	353
----------------------	-----

O

OTS_Monster::hasImmunity()	229
Checks if monster has given immunity.	
OTS_OTBMFile	229
OTBM format reader.	
OTS_OTBMFile::OTBM_ATTR_ACTION_ID	230
Action ID.	
OTS_Monster::getVoices()	228
Returns voices that monster can sound.	
OTS_Monster::getSpeed()	228
Returns monster speed.	
OTS_Monster::getName()	227
Returns monster name.	
OTS_Monster::getRace()	228
Returns monster race.	
OTS_OTBMFile::OTBM_ATTR_DEPOT_ID	230
Depot ID.	
OTS_OTBMFile::OTBM_ATTR_DESC	230
Description.	
OTS_OTBMFile::OTBM_ATTR_HOUSEDOORID	232
ID of doors.	
OTS_OTBMFile::OTBM_ATTR_ITEM	232
Item.	
OTS_OTBMFile::OTBM_ATTR_RUNE_CHARGES	233
Rune charges amount.	
OTS_OTBMFile::OTBM_ATTR_EXT_SPAWN_FILE	232
External spawns file.	
OTS_OTBMFile::OTBM_ATTR_EXT_HOUSE_FILE	231
External houses file.	
OTS_OTBMFile::OTBM_ATTR_DESCRIPTION	231
Description attribute.	
OTS_OTBMFile::OTBM_ATTR_EXT_FILE	231
External file.	
OTS_Monster::getManaCost()	227
Returns amount of mana required to summon this monster.	
OTS_Monster::getLoot()	227
Returns all possible loot.	
OTS_MapCoords::getZ()	222
Returns Z.	
OTS_MapCoords::set_state()	223
Magic PHP5 method.	
OTS_Monster	223
Wrapper for monsters files DOMDocument.	
OTS_MapCoords::getY()	222
Returns Y.	
OTS_MapCoords::getX()	222
Returns X.	
OTS_ItemType::set_state()	220
Magic PHP5 method.	
OTS_MapCoords	221
Map position point.	
OTS_Monster::getArmor()	224

<i>Returns monster armor.</i>	224
OTS Monster::getAttacks()	224
<i>Returns list of monster attacks.</i>	
OTS Monster::getFlags()	226
<i>Returns all monster flags (in format flagname => value).</i>	
OTS Monster::getHealth()	226
<i>Returns monster HP.</i>	
OTS Monster::getImmunities()	226
<i>Returns all monster immunities.</i>	
OTS Monster::getFlag()	225
<i>Returns specified flag value.</i>	
OTS Monster::getExperience()	225
<i>Returns amount of experience for killing this monster.</i>	
OTS Monster::getDefense()	224
<i>Returns monster defense rate.</i>	
OTS Monster::getDefenses()	225
<i>Returns list of special defenses.</i>	
OTS OTBMFile::OTBM_ATTR_TELE_DEST	233
<i>Teleport destination.</i>	
OTS OTBMFile::OTBM_ATTR_TEXT	233
<i>Text.</i>	
OTS OTBMFile::getTownsList()	241
<i>Returns list (id => name) of loaded towns.</i>	
OTS OTBMFile::getTownTemple()	242
<i>Returns town's temple position.</i>	
OTS OTBMFile::getWidth()	242
<i>Returns map width.</i>	
OTS OTBMFile::getTownName()	241
<i>Returns name of given town's ID.</i>	
OTS OTBMFile::getTownID()	241
<i>Returns town's ID.</i>	
OTS OTBMFile::getDescription()	240
<i>Returns map description.</i>	
OTS OTBMFile::getHeight()	240
<i>Returns map height.</i>	
OTS OTBMFile::key()	243
<i>Returns ID of current position.</i>	
OTS OTBMFile::loadFile()	243
<i>Loads OTBM file content.</i>	
OTS OTBMFile::__wakeup()	245
<i>Magic PHP5 method.</i>	
OTS Player	246
<i>OTServ character abstraction.</i>	
OTS Player::ban()	246
<i>Bans current player.</i>	
OTS OTBMFile::__set_state()	245
<i>Magic PHP5 method.</i>	
OTS OTBMFile::valid()	244
<i>Checks if there is anything more in iterator.</i>	
OTS OTBMFile::next()	243
<i>Moves to next iterator town.</i>	
OTS OTBMFile::rewind()	244
<i>Resets iterator index.</i>	

OTS OTBMFile::current()	239
<i>Returns town at current position in iterator.</i>	
OTS OTBMFile::count()	239
<i>Returns amount of towns loaded.</i>	
OTS OTBMFile::OTBM_NODE_ITEM_DEF	235
<i>Item definition.</i>	
OTS OTBMFile::OTBM_NODE_MAP_DATA	235
<i>Map data container.</i>	
OTS OTBMFile::OTBM_NODE_MONSTER	236
<i>Monster.</i>	
OTS OTBMFile::OTBM_NODE_ITEM	235
<i>Item.</i>	
OTS OTBMFile::OTBM_NODE_HOUSED_TILE	234
<i>Tile of house.</i>	
OTS OTBMFile::OTBM_ATTR_TILE_FLAGS	234
<i>Tile flags.</i>	
OTS OTBMFile::OTBM_ATTR_UNIQUE_ID	234
<i>Unique ID.</i>	
OTS OTBMFile::OTBM_NODE_ROOTV1	236
<i>Root node.</i>	
OTS OTBMFile::OTBM_NODE_SPAWNS	236
<i>Spawns container.</i>	
OTS OTBMFile::OTBM_NODE_TILE_SQUARE	238
<i>Tile.</i>	
OTS OTBMFile::OTBM_NODE_TOWN	238
<i>Town.</i>	
OTS OTBMFile::OTBM_NODE_TOWNS	239
<i>Towns container.</i>	
OTS OTBMFile::OTBM_NODE_TILE_REF	238
<i>Tile reference.</i>	
OTS OTBMFile::OTBM_NODE_TILE_AREA	237
<i>Map tiles fragment.</i>	
OTS OTBMFile::OTBM_NODE_SPAWN_AREA	237
<i>Spawn.</i>	
OTS OTBMFile::OTBM_NODE_TILE	237
<i>Single tile.</i>	
OTS ItemType::setType()	220
<i>Sets item type.</i>	
OTS ItemType::setName()	219
<i>Sets item type name.</i>	
OTS ItemType::FLAG_USEABLE	203
<i>Can be used by players.</i>	
OTS ItemType::FLAG_VERTICAL	203
<i>Is oriented vertically.</i>	
OTS ItemType::ITEM_GROUP_AMMUNITION	203
<i>Ammunition.</i>	
OTS ItemType::FLAG_STACKABLE	202
<i>Can be grouped with another items.</i>	
OTS ItemType::FLAG_ROTABLE	202
<i>Can be rotated by player.</i>	
OTS ItemType::FLAG_PICKUPABLE	201
<i>Can be picked up by player.</i>	
OTS ItemType::FLAG_READABLE	202

<i>Has readable sign.</i>	
OTS_ItemType::ITEM_GROUP_ARMOR	204
<i>Armor.</i>	
OTS_ItemType::ITEM_GROUP_CONTAINER	204
<i>Container.</i>	
OTS_ItemType::ITEM_GROUP_MAGICFIELD	206
<i>Magic field.</i>	
OTS_ItemType::ITEM_GROUP_NONE	206
<i>No group speciffied.</i>	
OTS_ItemType::ITEM_GROUP_RUNE	206
<i>Rune.</i>	
OTS_ItemType::ITEM_GROUP_KEY	205
<i>Key.</i>	
OTS_ItemType::ITEM_GROUP_GROUND	205
<i>Ground tile.</i>	
OTS_ItemType::ITEM_GROUP_DOOR	204
<i>Door.</i>	
OTS_ItemType::ITEM_GROUP_FLUID	205
<i>Liquid thing.</i>	
OTS_ItemType::FLAG_MOVEABLE	201
<i>Can be moved by player.</i>	
OTS_ItemType::FLAG_HORIZONTAL	201
<i>Is oriented horizontally.</i>	
OTS_ItemType::FLAG_ALWAYSONTOP	196
<i>Is always over other items in stack.</i>	
OTS_ItemType::FLAG_BLOCK_PATHFIND	197
<i>Can block searching for path.</i>	
OTS_ItemType::FLAG_BLOCK_PROJECTILE	197
<i>BLOCK_PROJECTILE flag(?).</i>	
OTS_ItemType::FLAG_ALLOWDISTREAD	196
<i>Can be read from distance.</i>	
OTS_ItemType	196
<i>Item type info.</i>	
OTS_ItemsList:: set_state()	195
<i>Magic PHP5 method.</i>	
OTS_ItemsList:: wakeup()	195
<i>Magic PHP5 method.</i>	
OTS_ItemType::FLAG_BLOCK_SOLID	197
<i>Can block characters from walking.</i>	
OTS_ItemType::FLAG_CANNOTDECAY	198
<i>Doesn't decay.</i>	
OTS_ItemType::FLAG_FLOORCHANGWEST	200
<i>Changes floor west from it's position.</i>	
OTS_ItemType::FLAG_HANGABLE	200
<i>Can be hang(?).</i>	
OTS_ItemType::FLAG_HAS_HEIGHT	200
<i>Does item rises stack height on it's field.</i>	
OTS_ItemType::FLAG_FLOORCHANGESOUTH	199
<i>Changes floor south from it's position.</i>	
OTS_ItemType::FLAG_FLOORCHANGENORTH	199
<i>Changes floor north from it's position.</i>	
OTS_ItemType::FLAG_FLOORCHANGEDOWN	198
<i>Changes floor under it.</i>	

OTS_ItemType::FLAG_FLOORCHANGE EAST	199
<i>Changes floor east from it's position.</i>	
OTS_ItemType::ITEM_GROUP_SPLASH	207
<i>Splash effect.</i>	
OTS_ItemType::ITEM_GROUP_TELEPORT	207
<i>Teleport field.</i>	
OTS_ItemType::isHorizontal()	215
<i>Checks if item is horizontal.</i>	
OTS_ItemType::isMovable()	215
<i>Checks if item is movable.</i>	
OTS_ItemType::isPickupable()	215
<i>Checks if item is pickupable.</i>	
OTS_ItemType::isHangable()	215
<i>Checks if item can be hanged.</i>	
OTS_ItemType::isBlocking()	214
<i>Checks if item is blocking.</i>	
OTS_ItemType::hasHeight()	214
<i>Checks if item has height.</i>	
OTS_ItemType::isAlwaysOnTop()	214
<i>Checks if item is always on top.</i>	
OTS_ItemType::isReadable()	216
<i>Checks if item is readable.</i>	
OTS_ItemType::isRotable()	216
<i>Checks if item can be rotated.</i>	
OTS_ItemType::setClientId()	218
<i>Sets client side ID.</i>	
OTS_ItemType::setFlags()	218
<i>Sets type flags.</i>	
OTS_ItemType::setGroup()	219
<i>Sets item group.</i>	
OTS_ItemType::setAttribute()	217
<i>Sets given attribute.</i>	
OTS_ItemType::isVertical()	217
<i>Checks if item is vertical.</i>	
OTS_ItemType::isStackable()	216
<i>Checks if item is stackable.</i>	
OTS_ItemType::isUsable()	217
<i>Checks if item is usable.</i>	
OTS_ItemType::getType()	213
<i>Returns item type.</i>	
OTS_ItemType::getName()	213
<i>Returns item name.</i>	
OTS_ItemType::ITEM_TYPE_DOOR	209
<i>Door.</i>	
OTS_ItemType::ITEM_TYPE_MAGICFIELD	209
<i>Magic field.</i>	
OTS_ItemType::ITEM_TYPE_MAILBOX	209
<i>Mailbox.</i>	
OTS_ItemType::ITEM_TYPE_DEPOT	208
<i>Depot locker.</i>	
OTS_ItemType::ITEM_TYPE_CONTAINER	208
<i>Container.</i>	
OTS_ItemType::ITEM_GROUP_WEAPON	207

<i>Weapon.</i>	
OTS_ItemType::ITEM_GROUP_WRITEABLE	208
<i>Item that can store editable sign.</i>	
OTS_ItemType::ITEM_TYPE_NONE	210
<i>No special type.</i>	
OTS_ItemType::ITEM_TYPE_TRASHHOLDER	210
<i>Trash can.</i>	
OTS_ItemType::getGroup()	212
<i>Returns group.</i>	
OTS_ItemType::getId()	213
<i>Returns item type server ID.</i>	
OTS_ItemType::getClientId()	212
<i>Returns item type client ID.</i>	
OTS_ItemType::getAttributesList()	212
<i>Returns all attributes list.</i>	
OTS_ItemType::createItem()	211
<i>Creates instance of this type.</i>	
OTS_ItemType::getAttribute()	211
<i>Returns given attribute.</i>	
OTS_Player::delete()	246
<i>Deletes player.</i>	
OTS_Player::find()	247
<i>Loads player by it's name.</i>	
OTS_Player::setTownId()	288
<i>Sets residence town's ID.</i>	
OTS_Player::setVocation()	289
<i>Sets player proffesion.</i>	
OTS_Player::unban()	289
<i>Deletes ban from current player.</i>	
OTS_Player::setSoul()	288
<i>Sets soul points.</i>	
OTS_Player::setSlot()	287
<i>Sets slot content.</i>	
OTS_Player::setSkill()	286
<i>Sets skill value.</i>	
OTS_Player::setSkillTries()	287
<i>Sets skill's tries for next level.</i>	
OTS_Player::unsetRedSkull()	290
<i>Unsets red skull flag.</i>	
OTS_Player::unsetSave()	290
<i>Unsets save flag.</i>	
OTS_Spell	292
<i>Wrapper for spell info.</i>	
OTS_Spell::getCharges()	293
<i>Number of rune charges.</i>	
OTS_Spell::getConjureCount()	293
<i>Returns amount of items conjured by this spell.</i>	
OTS_Players_List::init()	292
<i>Sets list parameters.</i>	
OTS_Players_List::deletePlayer()	291
<i>Deletes player.</i>	
OTS_Player::sleep()	290
<i>Magic PHP5 method.</i>	

OTS Players List	291
<i>List of players.</i>	
OTS Player::setSex()	286
<i>Sets player gender.</i>	
OTS Player::setSave()	285
<i>Sets save flag.</i>	
OTS Player::setMana()	280
<i>Sets current mana.</i>	
OTS Player::setManaMax()	280
<i>Sets maximum mana.</i>	
OTS Player::setManaSpent()	281
<i>Sets mana spent.</i>	
OTS Player::setMagLevel()	279
<i>Sets magic level.</i>	
OTS Player::setLossSkills()	279
<i>Sets percentage of skills lost after dead.</i>	
OTS Player::setLossExperience()	278
<i>Sets percentage of experience lost after dead.</i>	
OTS Player::setLossMana()	279
<i>Sets percentage of used mana lost after dead.</i>	
OTS Player::setName()	281
<i>Sets players's name.</i>	
OTS Player::setPosX()	282
<i>Sets X map coordinate.</i>	
OTS Player::setRankId()	284
<i>Sets guild rank ID.</i>	
OTS Player::setRedSkull()	284
<i>Sets red skull flag.</i>	
OTS Player::setRedSkullTime()	285
<i>Sets red skulled time remained.</i>	
OTS Player::setRank()	284
<i>Assigns guild rank.</i>	
OTS Player::setPremiumEnd()	283
<i>Sets player's Premium Account expiration timestamp.</i>	
OTS Player::setPosY()	282
<i>Sets Y map coordinate.</i>	
OTS Player::setPosZ()	283
<i>Sets Z map coordinate.</i>	
OTS Spell::getConjureId()	294
<i>Returns ID of item conjured by this spell.</i>	
OTS Spell::getID()	294
<i>Returns rune item id.</i>	
OTS SQLFilter::OPERATOR_GREATER	302
<i>Greater-than operator.</i>	
OTS SQLFilter::OPERATOR LIKE	303
<i>LIKE operator.</i>	
OTS SQLFilter::OPERATOR LOWER	303
<i>Lower-than operator.</i>	
OTS SQLFilter::OPERATOR EQUAL	302
<i>Equal operator.</i>	
OTS SQLFilter::CRITERIUM OR	302
<i>OR sibling.</i>	
OTS SQLFilter	301

SQL WHERE clause object.	301
OTS SQLFilter::CRITERIUM AND	301
AND sibling.	
OTS SQLFilter::OPERATOR NEQUAL	303
Not-equal operator.	
OTS SQLFilter::OPERATOR NGREATER	304
Not-greater-than operator.	
OTS SQLFilter::getTables()	306
Returns list of all tables used by filter.	
OTS SQLFilter::sleep()	306
Magic PHP5 method.	
OTS SQLFilter::toString()	307
Returns string representation of WHERE clause.	
OTS SQLFilter::compareField()	305
Compares field with a literal value.	
OTS SQLFilter::addFilter()	305
General-purpose filter.	
OTS SQLFilter::OPERATOR NLIKE	304
Not-LIKE operator.	
OTS SQLFilter::OPERATOR NLOWER	304
Not-lower-than operator.	
OTS SQLField::getTable()	300
Returns table name.	
OTS SQLField::getName()	300
Returns field name.	
OTS Spell::getReagentId()	296
Returns ID of item that is used by spell.	
OTS Spell::getSoul()	296
Soul points cost.	
OTS Spell::getType()	296
Returns spell type.	
OTS Spell::getName()	295
Returns spell name.	
OTS Spell::getMana()	295
Mana cost.	
OTS Spell::getLevel()	294
Level required for use.	
OTS Spell::getMagicLevel()	295
Magic level required to cast.	
OTS Spell::getVocations()	297
Returns list of vocations that are allowed to learn this spell.	
OTS Spell::getWords()	297
Returns spell formula.	
OTS Spell::isLearnNeeded()	299
Checks if spell needs to be learned.	
OTS Spell::isPremium()	299
Checks if spell requires PACC.	
OTS SQLField	299
SQL identifier representation.	
OTS Spell::isFarUseAllowed()	298
Checks if distance use allowed.	
OTS Spell::isEnabled()	298
Checks if spell is enabled.	

OTS Spell::hasParams()	297
<i>Checks if spell has parameter.</i>	
OTS Spell::isAggressive()	298
<i>Checks if spell is threatened as unfriendly by other creatures.</i>	
OTS Player::setLookType()	278
<i>Sets outfit.</i>	
OTS Player::setLookLegs()	277
<i>Sets legs color.</i>	
OTS Player::getLossMana()	256
<i>Percentage of used mana lost after dead.</i>	
OTS Player::getLossSkills()	257
<i>Percentage of skills lost after dead.</i>	
OTS Player::getMagLevel()	257
<i>Magic level.</i>	
OTS Player::getLossExperience()	256
<i>Percentage of experience lost after dead.</i>	
OTS Player::getLookType()	256
<i>Outfit.</i>	
OTS Player::getLookHead()	255
<i>Hair color.</i>	
OTS Player::getLookLegs()	255
<i>Legs color.</i>	
OTS Player::getMana()	258
<i>Current mana.</i>	
OTS Player::getManaMax()	258
<i>Maximum mana.</i>	
OTS Player::getPosZ()	260
<i>Z map coordinate.</i>	
OTS Player::getPremiumEnd()	260
<i>Player's Premium Account expiration timestamp.</i>	
OTS Player::getRank()	261
<i>Assigned guild rank.</i>	
OTS Player::getPosY()	260
<i>Y map coordinate.</i>	
OTS Player::getPosX()	259
<i>X map coordinate.</i>	
OTS Player::getManaSpent()	258
<i>Mana spent.</i>	
OTS Player::getName()	259
<i>Player name.</i>	
OTS Player::getLookFeet()	254
<i>Boots color.</i>	
OTS Player::getLookBody()	254
<i>Body color.</i>	
OTS Player::getDepot()	249
<i>Returns items tree from given depot.</i>	
OTS Player::getDirection()	250
<i>Looking direction.</i>	
OTS Player::getExperience()	250
<i>Experience points.</i>	
OTS Player::getCustomField()	249
<i>Reads custom field.</i>	
OTS Player::getConditions()	248

<i>Conditions.</i>	
OTS_Player::getAccount()	247
<i>Returns account of this player.</i>	
OTS_Player::getCap()	248
<i>Capacity.</i>	
OTS_Player::getGroup()	251
<i>Returns group of this player.</i>	
OTS_Player::getGuildNick()	251
<i>Guild nick.</i>	
OTS_Player::getLastLogin()	253
<i>Last login timestamp.</i>	
OTS_Player::getLevel()	253
<i>Experience level.</i>	
OTS_Player::getLookAddons()	254
<i>Addons.</i>	
OTS_Player::getLastIP()	253
<i>Last login IP.</i>	
OTS_Player::getId()	252
<i>Player ID.</i>	
OTS_Player::getHealth()	251
<i>Current HP.</i>	
OTS_Player::getHealthMax()	252
<i>Maximum HP.</i>	
OTS_Player::getRankId()	261
<i>Guild rank ID.</i>	
OTS_Player::getRedSkullTime()	262
<i>Red skulled time remained.</i>	
OTS_Player::setExperience()	272
<i>Sets experience points.</i>	
OTS_Player::setGroup()	272
<i>Assigns character to group.</i>	
OTS_Player::setGuildNick()	273
<i>Sets guild nick.</i>	
OTS_Player::setDirection()	271
<i>Sets looking direction.</i>	
OTS_Player::setDepot()	271
<i>Sets depot content.</i>	
OTS_Player::setConditions()	269
<i>Sets conditions.</i>	
OTS_Player::setCustomField()	270
<i>Writes custom field.</i>	
OTS_Player::setHealth()	273
<i>Sets current HP.</i>	
OTS_Player::setHealthMax()	274
<i>Sets maximum HP.</i>	
OTS_Player::setLookBody()	276
<i>Sets body color.</i>	
OTS_Player::setLookFeet()	276
<i>Sets boots color.</i>	
OTS_Player::setLookHead()	277
<i>Sets hair color.</i>	
OTS_Player::setLookAddons()	275
<i>Sets addons.</i>	

OTS_Player::setLevel()	275
<i>Sets experience level.</i>	
OTS_Player::setLastIP()	274
<i>Sets last login IP.</i>	
OTS_Player::setLastLogin()	274
<i>Sets last login timestamp.</i>	
OTS_Player::setCap()	269
<i>Sets capacity.</i>	
OTS_Player::setAccount()	268
<i>Assigns character to account.</i>	
OTS_Player::getSlot()	264
<i>Returns items tree from given slot.</i>	
OTS_Player::getSoul()	265
<i>Soul points.</i>	
OTS_Player::getTownId()	265
<i>Residence town's ID.</i>	
OTS_Player::getSkillTries()	263
<i>Returns player's skill's tries for next level.</i>	
OTS_Player::getSkill()	263
<i>Returns player's skill.</i>	
OTS_Player::getSave()	262
<i>Save counter.</i>	
OTS_Player::getSex()	262
<i>Player gender.</i>	
OTS_Player::getVocation()	265
<i>Player proffesion.</i>	
OTS_Player::getVocationName()	266
<i>Player proffesion name.</i>	
OTS_Player::load()	268
<i>Loads player with given id.</i>	
OTS_Player::save()	268
<i>Saves player in database.</i>	
OTS_Player::isSaveSet()	267
<i>Checks if save flag is set.</i>	
OTS_Player::isLoading()	267
<i>Checks if object is loaded.</i>	
OTS_Player::hasRedSkull()	266
<i>Checks if player has red skull.</i>	
OTS_Player::isBanned()	267
<i>Checks if player is banned.</i>	
OTS_ItemsList::valid()	194
<i>Checks if there is anything more in interator.</i>	
OTS_ItemsList::rewind()	194
<i>Resets iterator index.</i>	
OTS_Container::rewind()	112
<i>Resets internal items array pointer.</i>	
OTS_Container::valid()	112
<i>Checks if there are any items left.</i>	
OTS_DB_MySQL	113
<i>MySQL connection interface.</i>	
OTS_Container::removeItem()	111
<i>Removes given item from current container.</i>	
OTS_Container::next()	111

OTS Container::current()	Moves to next item.	110
OTS Container::key()	Returns current item.	111
OTS DB MySQL::fieldName()	Current cursor position.	114
OTS DB MySQL::limit()	Query-quoted field name.	114
OTS DB ODBC::fieldName()	LIMIT/OFFSET clause for queries.	117
OTS DB ODBC::limit()	Query-quoted field name.	118
OTS DB ODBC::SQLquery()	LIMIT/OFFSET clause for queries.	118
OTS DB ODBC	IOTS_DB method.	116
OTS DB MySQL::tableName()	ODBC connection interface.	116
OTS DB MySQL::SQLquery()	Query-quoted table name.	115
OTS DB MySQL::SQLquote()	IOTS_DB method.	115
OTS Container::count()	IOTS_DB method.	110
OTS Container::addItem()	Number of items inside container.	109
OTS Base List::next()	Adds item to container.	105
OTS Base List::orderBy()	Moves to next row.	105
OTS Base List::resetOrder()	Appends sorting rule.	105
OTS Base List::key()	Clears ORDER BY clause.	104
OTS Base List::init()	Current cursor position.	104
OTS Base List::count()	Sets list parameters.	103
OTS Base List::current()	Returns number of accounts on list in current criterium.	104
OTS Base List::rewind()	Returns current row.	106
OTS Base List::setFilter()	Select rows from database.	106
OTS Base List::sleep()	Sets filter on list.	108
OTS Base List::wakeup()	Magic PHP5 method.	109
OTS Container	Magic PHP5 method.	109
OTS Base List::set_state()	Container item representation.	108
	Magic PHP5 method.	

OTS_Base_List::valid()	107
<i>Checks if there are any rows left.</i>	
OTS_Base_List::setLimit()	107
<i>Sets LIMIT.</i>	
OTS_Base_List::setOffset()	107
<i>Sets OFFSET.</i>	
OTS_DB_ODBC::SQLquote()	119
<i>IOTS_DB method.</i>	
OTS_DB_ODBC::tableName()	119
<i>Query-quoted table name.</i>	
OTS_FileLoader::sleep()	131
<i>Magic PHP5 method.</i>	
OTS_FileNode	132
<i>OTServ binary file node representation.</i>	
OTS_FileNode::getBuffer()	132
<i>Returns properties stream.</i>	
OTS_FileLoader::set_state()	131
<i>Magic PHP5 method.</i>	
OTS_FileLoader::clone()	130
<i>Creates clone of object.</i>	
OTS_FileLoader::loadFile()	129
<i>Opens file.</i>	
OTS_FileLoader::setCacheDriver()	130
<i>Sets cache handler.</i>	
OTS_FileNode::getChar()	132
<i>Returns single byte.</i>	
OTS_FileNode::getChild()	133
<i>Returns first child.</i>	
OTS_FileNode::getType()	134
<i>Returns node type.</i>	
OTS_FileNode::isValid()	135
<i>Checks if there is anything left in stream.</i>	
OTS_FileNode::setBuffer()	135
<i>Sets properties stream.</i>	
OTS_FileNode::getString()	134
<i>Returns string from buffer.</i>	
OTS_FileNode::getShort()	134
<i>Returns double byte.</i>	
OTS_FileNode::getLong()	133
<i>Returns quater byte.</i>	
OTS_FileNode::getNext()	133
<i>Returns next sibling.</i>	
OTS_FileLoader::\$root	129
<i>Root node.</i>	
OTS_FileLoader::\$cache	128
<i>Cache handler.</i>	
OTS_DB_PostgreSQL::SQLquote()	122
<i>IOTS_DB method.</i>	
OTS_DB_PostgreSQL::tableName()	123
<i>Query-quoted table name.</i>	
OTS_DB_SQLite	124
<i>SQLite connection interface.</i>	
OTS_DB_PostgreSQL::SQLquery()	122

IOTS_DB method.	
OTS_DB PostgreSQL::limit()	121
<i>LIMIT/OFFSET clause for queries.</i>	
OTS_DB PostgreSQL	120
<i>PostgreSQL connection interface.</i>	
OTS_DB PostgreSQL::fieldName()	121
<i>Query-quoted field name.</i>	
OTS_DB SQLite::fieldName()	124
<i>Query-quoted field name.</i>	
OTS_DB SQLite::limit()	125
<i>LIMIT/OFFSET clause for queries.</i>	
OTS FileLoader::ESCAPE_CHAR	127
<i>Escape another special byte.</i>	
OTS FileLoader::NODE_END	128
<i>End of node.</i>	
OTS FileLoader::NODE_START	128
<i>Start of node.</i>	
OTS FileLoader	127
<i>Universal OTServ binary formats reader.</i>	
OTS_DB SQLite::tableName()	126
<i>Query-quoted table name.</i>	
OTS_DB SQLite::SQLquery()	125
<i>IOTS_DB method.</i>	
OTS_DB SQLite::SQLquote()	126
<i>IOTS_DB method.</i>	
OTS Base_List::\$table	102
<i>Default table name for queries.</i>	
OTS Base_List::\$class	102
<i>Class of generated objects.</i>	
OTS Monster.php	59
OTS OTBMFile.php	60
<i>Code in this file bases on oryiginal OTServ OTBM format loading C++ code (iomapotbm.</i>	
OTS Player.php	61
OTS MapCoords.php	58
<i>Code in this file bases on oryiginal OTServ OTBM format loading C++ code (iomapotbm.</i>	
OTS ItemType.php	57
<i>Code in this file bases on oryiginal OTServ items loading C++ code (items.</i>	
OTS Item.php	55
OTS ItemsList.php	56
<i>Code in this file bases on oryiginal OTServ items loading C++ code (itemloader.</i>	
OTS Players_List.php	62
OTS Spell.php	63
OTS Account::ban()	84
<i>Bans current account.</i>	
OTS Account::block()	84
<i>Blocks account.</i>	
OTS Account::count()	85
<i>Returns number of player within.</i>	
OTS Account	83
<i>OTServ account abstraction.</i>	
OTS SQLite_Results.php	66
OTS SQLiteField.php	64
OTS SQLiteFilter.php	65

OTS_InfoRespond.php	54
OTS_Guilds_List.php	53
OTS_Container.php	41
OTS_DB_MySQL.php	42
OTS_DB_ODBC.php	43
OTS_Base_List.php	40
OTS_Base_DAO.php	39
OTS_Account.php	37
OTS_Accounts_List.php	38
OTS_DB_PostgreSQL.php	44
OTS_DB_SQLite.php	45
OTS_Guild.php	50
OTS_GuildRank.php	51
OTS_GuildRanks_List.php	52
OTS_Groups_List.php	49
OTS_Group.php	48
OTS_FileLoader.php	46
Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader.	
OTS_FileNode.php	47
Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader.	
OTS_Account::create()	85
Creates new account.	
OTS_Account::createEx()	86
Creates new account.	
OTS_Account::unban()	97
Deletes ban from current account.	
OTS_Account::unblock()	97
Unblocks account.	
OTS_Accounts_List	97
List of accounts.	
OTS_Account::setPassword()	96
Sets account's password.	
OTS_Account::setPACCDays()	96
Sets PACC days count.	
OTS_Account::setEMail()	95
Sets account's email.	
OTS_Account::setGroup()	95
Assigns account to group.	
OTS_Accounts_List::deleteAccount()	98
Deletes account.	
OTS_Accounts_List::init()	98
Sets list parameters.	
OTS_Base_DAO::sleep()	101
Magic PHP5 method.	
OTS_Base_DAO::wakeup()	101
Magic PHP5 method.	
OTS_Base_List	102
Basic list class routines.	
OTS_Base_DAO::set_state()	100
Magic PHP5 method.	
OTS_Base_DAO::clone()	100
Creates clone of object.	
OTS_Base_DAO	99

<i>Basic data access object routines.</i>	
OTS_Base.DAO::\$db	99
<i>Database connection.</i>	
OTS_Account::setCustomField()	94
<i>Writes custom field.</i>	
OTS_Account::save()	93
<i>Updates account in database.</i>	
OTS_Account::getGroup()	89
<i>Returns group of this account.</i>	
OTS_Account::getId()	89
<i>Account number.</i>	
OTS_Account::getIterator()	90
<i>Returns players iterator.</i>	
OTS_Account::getEmail()	88
<i>E-mail address.</i>	
OTS_Account::getCustomField()	88
<i>Reads custom field.</i>	
OTS_Account::delete()	87
<i>Deletes account.</i>	
OTS_Account::find()	87
<i>Loads account by it's e-mail address.</i>	
OTS_Account::getPACCDays()	90
<i>PACC days.</i>	
OTS_Account::getPassword()	91
<i>Account's password.</i>	
OTS_Account::isLoading()	93
<i>Checks if object is loaded.</i>	
OTS_Account::load()	93
<i>Loads account with given number.</i>	
OTS_Account::isBlocked()	92
<i>Checks if account is blocked.</i>	
OTS_Account::isBanned()	92
<i>Checks if account is banned.</i>	
OTS_Account::getPlayers()	91
<i>List of characters on account.</i>	
OTS_Account::getPlayersList()	91
<i>List of characters on account.</i>	
OTS_FileNode::setChild()	135
<i>Sets first child.</i>	
OTS_FileNode::setNext()	136
<i>Sets next sibling.</i>	
OTS_InfoRespond::getMapWidth()	177
<i>Returns map width.</i>	
OTS_InfoRespond::getMaxPlayers()	178
<i>Returns maximum amount of players online.</i>	
OTS_InfoRespond::getMonstersCount()	178
<i>Returns number of all monsters on map.</i>	
OTS_InfoRespond::getMapName()	177
<i>Returns map name.</i>	
OTS_InfoRespond::getMapHeight()	177
<i>Returns map height.</i>	
OTS_InfoRespond::getLocation()	176
<i>Returns server location.</i>	

OTS_InfoRespond::getMapAuthor()	176
<i>Returns map author.</i>	
OTS_InfoRespond::getMOTD()	178
<i>Returns server's Message Of The Day</i>	
OTS_InfoRespond::getName()	178
<i>Returns server name.</i>	
OTS_InfoRespond::getServer()	180
<i>Returns server attribute.</i>	
OTS_InfoRespond::getServerVersion()	180
<i>Returns server version.</i>	
OTS_InfoRespond::getTSPQVersion()	181
<i>Returns version of root element.</i>	
OTS_InfoRespond::getPort()	180
<i>Returns server port.</i>	
OTS_InfoRespond::getPlayersPeak()	179
<i>Returns record of online players.</i>	
OTS_InfoRespond::getOnlinePlayers()	179
<i>Returns current amount of players online.</i>	
OTS_InfoRespond::getOwner()	179
<i>Returns owner name.</i>	
OTS_InfoRespond::getIP()	176
<i>Returns server IP.</i>	
OTS_InfoRespond::getEmail()	175
<i>Returns owner e-mail.</i>	
OTS_GuildRank::setCustomField()	170
<i>Writes custom field.</i>	
OTS_GuildRank::setGuild()	170
<i>Assigns rank to guild.</i>	
OTS_GuildRank::setLevel()	171
<i>Sets rank's access level within guild.</i>	
OTS_GuildRank::save()	169
<i>Saves rank in database.</i>	
OTS_GuildRank::load()	169
<i>Loads rank with given id.</i>	
OTS_GuildRank::getPlayersList()	168
<i>List of characters with current rank.</i>	
OTS_GuildRank::isLoading()	169
<i>Checks if object is loaded.</i>	
OTS_GuildRank::setName()	171
<i>Sets rank's name.</i>	
OTS_GuildRanks_List	172
<i>List of guild ranks.</i>	
OTS_Guilds_List::init()	174
<i>Sets list parameters.</i>	
OTS_InfoRespond	175
<i>Wrapper for 'info' respond's DOMDocument.</i>	
OTS_InfoRespond::getClientVersion()	175
<i>Returns dedicated version of client.</i>	
OTS_Guilds_List::deleteGuild()	174
<i>Deletes guild.</i>	
OTS_Guilds_List	173
<i>List of guilds.</i>	
OTS_GuildRanks_List::deleteGuildRank()	172

<i>Deletes guild rank.</i>	
OTS_GuildRanks_List::init()	173
<i>Sets list parameters.</i>	
OTS_InfoRespond::getUptime()	181
<i>Returns server uptime.</i>	
OTS_InfoRespond::getURL()	181
<i>Returns server website.</i>	
OTS_ItemsList::ROOT_ATTR_VERSION	190
<i>Root file attribute.</i>	
OTS_ItemsList::count()	190
<i>Returns amount of items loaded.</i>	
OTS_ItemsList::current()	190
<i>Returns item at current position in iterator.</i>	
OTS_ItemsList::ITEM_ATTR_TOPORDER	189
<i>Always-on-top order.</i>	
OTS_ItemsList::ITEM_ATTR_SPEED	189
<i>Speed.</i>	
OTS_ItemsList::ITEM_ATTR_LIGHT2	188
<i>Light.</i>	
OTS_ItemsList::ITEM_ATTR_SERVERID	189
<i>Server ID.</i>	
OTS_ItemsList::getBuildVersion()	191
<i>Returns build version.</i>	
OTS_ItemsList::getClientVersion()	191
<i>Returns client version.</i>	
OTS_ItemsList::key()	193
<i>Returns ID of current position.</i>	
OTS_ItemsList::loadItems()	193
<i>Loads items.</i>	
OTS_ItemsList::next()	194
<i>Moves to next iterator item.</i>	
OTS_ItemsList::getOTBVersion()	193
<i>Returns OTB file version.</i>	
OTS_ItemsList::getItemTypesList()	192
<i>Returns all loaded items.</i>	
OTS_ItemsList::getItemType()	191
<i>Returns given item type.</i>	
OTS_ItemsList::getItemTypeId()	192
<i>Finds item type by it's name.</i>	
OTS_ItemsList::ITEM_ATTR_CLIENTID	188
<i>Client ID.</i>	
OTS_ItemsList::CLIENT_VERSION_800	188
<i>Tibia client 8.</i>	
OTS_Item::getId()	184
<i>Returns item type.</i>	
OTS_Item::setAttributes()	184
<i>Sets item attributes.</i>	
OTS_Item::setCount()	184
<i>Sets count of item.</i>	
OTS_Item::getCount()	183
<i>Returns count of item.</i>	
OTS_Item::getAttributes()	183
<i>Returns item custom attributes.</i>	

OTS_Item	Single item representation.	182
OTS_Item::count()	Count value for current item.	183
OTS_ItemsList	Items list loader.	185
OTS_ItemsList::CLIENT_VERSION_750	Tibia client 7.	185
OTS_ItemsList::CLIENT_VERSION_790	Tibia client 7.	187
OTS_ItemsList::CLIENT_VERSION_792	Tibia client 7.	187
OTS_ItemsList::CLIENT_VERSION_780	Tibia client 7.	187
OTS_ItemsList::CLIENT_VERSION_770	Tibia client 7.	186
OTS_ItemsList::CLIENT_VERSION_755	Tibia client 7.	186
OTS_ItemsList::CLIENT_VERSION_760	Tibia client 7.	186
OTS_GuildRank::getPlayers()	Reads all players who has this rank set.	168
OTS_GuildRank::getName()	Rank name.	167
OTS_Group::setFlags()	Sets rights flags.	146
OTS_Group::setMaxDepotItems()	Sets maximum count of items in depot.	147
OTS_Group::setMaxVIPList()	Sets maximum count of players in VIP list.	147
OTS_Group::setCustomField()	Writes custom field.	145
OTS_Group::setAccess()	Sets access level.	145
OTS_Group::load()	Loads group with given id.	144
OTS_Group::save()	Saves account in database.	145
OTS_Group::setName()	Sets group's name.	148
OTS_Groups_List	List of groups.	148
OTS_Guild::acceptRequest()	Accepts player.	150
OTS_Guild::count()	Returns number of ranks within.	151
OTS_Guild::delete()	Deletes guild.	151
OTS_Guild::acceptInvite()	Finalise invitation.	150
OTS_Guild	OTServ guild abstraction.	149
OTS_Groups_List::deleteGroup()		148

<i>Deletes group.</i>	
OTS_Groups_List::init()	149
<i>Sets list parameters.</i>	
OTS_Group::isLoading()	144
<i>Checks if object is loaded.</i>	
OTS_Group::getPlayersList()	143
<i>List of characters in group.</i>	
OTS_Group	138
<i>OTServ user group abstraction.</i>	
OTS_Group::count()	139
<i>Returns number of player within.</i>	
OTS_Group::delete()	139
<i>Deletes group.</i>	
OTS_FileNode::__set_state()	138
<i>Magic PHP5 method.</i>	
OTS_FileNode::__clone()	137
<i>Creates clone of object.</i>	
OTS_FileNode::setType()	136
<i>Sets node type.</i>	
OTS_FileNode::skip()	137
<i>Skips given amount of bytes.</i>	
OTS_Group::getAccess()	139
<i>Access level.</i>	
OTS_Group::getCustomField()	140
<i>Reads custom field.</i>	
OTS_Group::getMaxVIPList()	142
<i>Maximum count of players in VIP list.</i>	
OTS_Group::getName()	142
<i>Group name.</i>	
OTS_Group::getPlayers()	143
<i>List of characters in given group.</i>	
OTS_Group::getMaxDepotItems()	142
<i>Maximum count of items in depot.</i>	
OTS_Group::getIterator()	141
<i>Returns players iterator.</i>	
OTS_Group::getFlags()	140
<i>Rights flags.</i>	
OTS_Group::getId()	141
<i>Group ID.</i>	
OTS_Guild::deleteInvite()	152
<i>Deletes invitation for player to guild.</i>	
OTS_Guild::deleteRequest()	152
<i>Deletes request from player.</i>	
OTS_Guild::setRequestsDriver()	162
<i>Assigns requests handler.</i>	
OTS_Guild::__clone()	163
<i>Creates clone of object.</i>	
OTS_Guild::__sleep()	163
<i>Magic PHP5 method.</i>	
OTS_Guild::setOwner()	162
<i>Assigns guild to owner.</i>	
OTS_Guild::setName()	161
<i>Sets players's name.</i>	

OTS_Guild::setCustomField()	160
<i>Writes custom field.</i>	
OTS_Guild::setInvitesDriver()	161
<i>Assigns invites handler.</i>	
OTS_GuildRank	163
<i>OTServ guild rank abstraction.</i>	
OTS_GuildRank::count()	164
<i>Returns number of player within.</i>	
OTS_GuildRank::getId()	166
<i>Rank ID.</i>	
OTS_GuildRank::getIterator()	166
<i>Returns players iterator.</i>	
OTS_GuildRank::getLevel()	167
<i>Rank's access level.</i>	
OTS_GuildRank::getGuild()	166
<i>Returns guild of this rank.</i>	
OTS_GuildRank::getCustomField()	165
<i>Reads custom field.</i>	
OTS_GuildRank::delete()	164
<i>Deletes guild rank.</i>	
OTS_GuildRank::find()	165
<i>Loads rank by it's name.</i>	
OTS_Guild::setCreationData()	160
<i>Sets guild creation data.</i>	
OTS_Guild::save()	159
<i>Saves guild in database.</i>	
OTS_Guild::getGuildRanksList()	155
<i>List of ranks in guild.</i>	
OTS_Guild::getId()	155
<i>Guild ID.</i>	
OTS_Guild::getIterator()	155
<i>Returns ranks iterator.</i>	
OTS_Guild::getGuildRanks()	154
<i>Reads all ranks that are in this guild.</i>	
OTS_Guild::getCustomField()	154
<i>Reads custom field.</i>	
OTS_Guild::find()	153
<i>Loads guild by it's name.</i>	
OTS_Guild::getCreationData()	153
<i>Guild creation data.</i>	
OTS_Guild::getName()	156
<i>Guild name.</i>	
OTS_Guild::getOwner()	156
<i>Returns owning player of this player.</i>	
OTS_Guild::load()	158
<i>Loads guild with given id.</i>	
OTS_Guild::request()	159
<i>Requests membership in guild for player player.</i>	
OTS_Guild::listRequests()	158
<i>Returns list of players that requested membership.</i>	
OTS_Guild::listInvites()	157
<i>Returns list of invited players.</i>	
OTS_Guild::invite()	157

<i>Invites player to guild.</i>	
OTS_Guild::isLoaded()	157
<i>Checks if object is loaded.</i>	
OTS.php	36
<i>This file contains main toolkit class.</i>	

P

POT::createFilter()	325
<i>Creates lists filter.</i>	
POT::connect()	324
<i>Connects to database.</i>	
POT::banIP()	323
<i>Bans given IP number.</i>	
POT::createObject()	326
<i>Creates OTServ DAO class instance.</i>	
POT::getConjure()	326
<i>Returns given conjure spell.</i>	
POT::getDBHandle()	327
<i>Returns database connection handle.</i>	
POT::getConjuresList()	327
<i>Returns list of conjure spells.</i>	
POT::VOCATION_SORCERER	323
<i>Sorcerer.</i>	
POT::VOCATION_PALADIN	323
<i>Paladin.</i>	
POT::SPELL_INSTANT	321
<i>Instant spell.</i>	
POT::SPELL_CONJURE	320
<i>Conjure spell.</i>	
POT::SLOT_RING	320
<i>Ring slot.</i>	
POT::SPELL_RUNE	321
<i>Rune spell.</i>	
POT::VOCATION_DRUID	322
<i>Druid.</i>	
POT::VOCATION_NONE	322
<i>None vocation.</i>	
POT::VOCATION_KNIGHT	322
<i>Knight.</i>	
POT::getInstance()	328
<i>Singleton.</i>	
POT::getInstant()	328
<i>Returns given instant spell.</i>	
POT::loadSpells()	334
<i>Loads spells list.</i>	
POT::loadMonsters()	333
<i>Loads monsters mapping file.</i>	
POT::loadClass()	333
<i>Loads POT class file.</i>	
POT::loadVocations()	334
<i>Loads vocations list.</i>	

POT::serverStatus()	335
<i>Queries server status.</i>	
POT::unbanIP()	337
<i>Deletes ban from given IP number.</i>	
POT::setPOTPath()	336
<i>Set POT directory.</i>	
POT::isIPBanned()	332
<i>Checks if given IP is banned.</i>	
POT::getVocationsList()	332
<i>Returns list (id => name) of loaded vocations.</i>	
POT::getMonstersList()	329
<i>Returns list of loaded monsters.</i>	
POT::getMonster()	329
<i>Returns loaded data of given monster.</i>	
POT::getInstantsList()	329
<i>Returns list of instants.</i>	
POT::getRune()	330
<i>Returns given rune spell.</i>	
POT::getRunesList()	330
<i>Returns list of runes.</i>	
POT::getVocationName()	331
<i>Returns name of given vocation's ID.</i>	
POT::getVocationID()	331
<i>Returns vocation's ID.</i>	
POT::SLOT_RIGHT	320
<i>Right hand slot.</i>	
POT::SLOT_NECKLACE	319
<i>Necklace slot.</i>	
POT::DIRECTION_EAST	311
<i>East.</i>	
POT::DEPOT_SID_FIRST	310
<i>First depot item sid.</i>	
POT::DB_SQLITE	310
<i>SQLite driver.</i>	
POT::DIRECTION_NORTH	311
<i>North.</i>	
POT::DIRECTION_SOUTH	311
<i>South.</i>	
POT::ORDER_ASC	312
<i>Ascending sorting order.</i>	
POT::DIRECTION_WEST	312
<i>West.</i>	
POT::DB_PGSQL	309
<i>PostgreSQL driver.</i>	
POT::DB_ODBC	309
<i>ODBC driver.</i>	
POT	307
<i>Main POT class.</i>	
POT class preview	5
PHP 5.0	3
POT::BAN_ACCOUNT	307
<i>Account ban.</i>	
POT::BAN_IP	308

<i>IP ban.</i>	309
POT::DB_MYSQL	308
<i>MySQL driver.</i>	
POT::BAN_PLAYER	312
<i>Player ban.</i>	
POT::ORDER_DESC	313
<i>Descending sorting order.</i>	
POT::SEX_FEMALE	317
<i>Female gender.</i>	
POT::SLOT_BACKPACK	317
<i>Backpack slot.</i>	
POT::SLOT_ARMOR	316
<i>Armor slot.</i>	
POT::SLOT_AMMO	318
<i>Ammunition slot.</i>	
POT::SLOT_FEET	318
<i>Boots slot.</i>	
POT::SLOT_HEAD	319
<i>Head slot.</i>	
POT::SLOT_LEGS	318
<i>Legs slot.</i>	
POT::SLOT_LEFT	316
<i>Left hand slot.</i>	
POT::SKILL_SWORD	315
<i>Sword fighting.</i>	
POT::SKILL_SHIELDING	313
<i>Shielding.</i>	
POT::SKILL_AXE	314
<i>Axe fighting.</i>	
POT::SEX_MALE	314
<i>Male gender.</i>	
POT::SKILL_CLUB	315
<i>Club fighting.</i>	
POT::SKILL_DISTANCE	315
<i>Distance fighting.</i>	
POT::SKILL_FIST	315
<i>Fist fighting.</i>	
POT::SKILL_FISHING	1
<i>Fishing.</i>	
POT	

Q

Quick start	6
-----------------------------	---

R

README	351
RULES	350

S

Server online status	19
--	----