

phpALUGen Documentation



Contents

Package phpALUGen Procedural Elements	2
keyTools.php	2
miscClasses.php	3
phpALUGen.php	4
productLibrary.php	5
Package phpALUGen Classes	6
Class phpALUGen	6
Method genKey	6
Method genProductKeys	6
Class phpALUGen License	7
Var \$Expiration	7
Var \$LicenseClass	7
Var \$Licensee	7
Var \$LicenseKey	7
Var \$LicenseType	8
Var \$MaxCount	8
Var \$ProductKey	8
Var \$ProductName	8
Var \$ProductVer	8
Var \$RegisteredDate	8
Method save	8
Method toString	8
Class phpALUGen ProductInfo	9
Var \$GCode	9
Var \$Name	9
Var \$VCode	9
Var \$Version	9
Class phpALUGen ProductLibrary	9
Method addProduct	10
Method getAllProducts	10
Method getProductsIni	10
Method loadProductsIni	10
Method removeProduct	10
Method retrieveProduct	11
Class phpALUGen ProductLibrary file	11
Constructor phpALUGen ProductLibrary file	11
Class phpALUGen ProductLibrary mysql	12
Constructor phpALUGen ProductLibrary mysql	12
Appendices	13
Appendix A - Class Trees	14
phpALUGen	14

Package phpALUGen Procedural Elements

keyTools.php

The file for **key** generation and manipulation functions and classes.

- **Package** phpALUGen
- **Copyright** Copyright (C) 2005 Andy Schmitz. Licensed under the GNU Lesser General Public License version 2.1
- **Author** Andy Schmitz < andy.schmitz@gmail.com >

miscClasses.php

The file for miscellaneous classes and definitions.

- **Package** phpALUGen
- **Copyright** Copyright (C) 2005 Andy Schmitz. Licensed under the GNU Lesser General Public License version 2.1
- **Author** Andy Schmitz < andy.schmitz@gmail.com >

phpALUGen.php

The main include file for the phpALUGen library.

Include this file, and none of the others in your program files. This file will create a product library object in `$phpALUGen_ProductLibrary` for ease of use and minimum load on databases and disks. Please use this object instead of creating your own.

- **Package** phpALUGen
- **Copyright** Copyright (C) 2005 Andy Schmitz. Licensed under the GNU Lesser General Public License version 2.1
- **Author** Andy Schmitz < andy.schmitz@gmail.com >

```
include $phpALUGen_dir.'/productLibrary.php' [line 59]
```

```
include $phpALUGen_dir.'/miscClasses.php' [line 61]
```

```
include $phpALUGen_dir.'/keyTools.php' [line 60]
```

```
include $configfile [line 51]
```

```
require_once 'Crypt/RSA.php' [line 64]
```

productLibrary.php

The file for product library functions and classes

- **Package** phpALUGen
- **Copyright** Copyright (C) 2005 Andy Schmitz. Licensed under the GNU Lesser General Public License version 2.1
- **Author** Andy Schmitz < andy.schmitz@gmail.com >

Package phpALUGen Classes

Class phpALUGen [line 37]

The main phpALUGen class.

- **Package** phpALUGen

string function phpALUGen::genKey(&\$lic, \$installCode) [line 50]

Function Parameters:

- [*phpALUGen License*](#) **\$lic** The license to generate the liberation key for. Must have ProductName and ProductVer set.
- *string* **\$installCode** The install code to generate the liberation key for. Should be provided by the user.

A function to generate liberation keys.

Takes an input license and install code and returns a liberation key. This is probably the slowest function in the entire library. Avoid calling it unnecessarily.

function phpALUGen::genProductKeys(\$bits, &\$VCode, &\$GCode) [line 228]

Function Parameters:

- *integer* **\$bits** The key length (in bits)
- *string* **\$VCode** The public key data (pass by reference, will be set by this function)
- *string* **\$GCode** The private key data (pass by reference, will be set by this function)

Generates new product keys (to be used in a new product) of a given number of bits.

Class phpALUGen_License

[line 37]

A class for handling license information.

- **Package** phpALUGen

phpALUGen_License::\$Expiration

string = [line 78]

The expiration date of the license (YYYY/MM/DD)

phpALUGen_License::\$LicenseClass

string = [line 84]

The class of the license (Single user or Multiple)

phpALUGen_License::\$Licensee

string = [line 54]

The user the license is for

phpALUGen_License::\$LicenseKey

string = [line 66]

The license key (RSA signature)

phpALUGen_License::\$LicenseType

integer = [line 90]

The type of the license (Periodic (1), Permanent (2), or Time Locked (3))

`phpALUGen_License::$MaxCount`

integer = [line 96]

The maximum number of times the product can be used

`phpALUGen_License::$ProductKey`

string = [line 60]

The VCode of the product (Public Key)

`phpALUGen_License::$ProductName`

string = [line 42]

The name of the product the license is for

`phpALUGen_License::$ProductVer`

string = [line 48]

The version of the product the license is for

`phpALUGen_License::$RegisteredDate`

string = [line 72]

The date the license was generated (YYYY/MM/DD)

string function phpALUGen_License::save() [line 124]

Generates a copy of this license to use for a liberation key.

This requires LicenseKey to be set.

string function phpALUGen_License::toString() [line 103]

Generates a copy of the license's information in a string.

Class phpALUGen_ProductInfo

[line 137]

A class for storing product information.

- **Package** phpALUGen

phpALUGen_ProductInfo::\$GCode

string = [line 154]

The GCode for the product (Private Key)

phpALUGen_ProductInfo::\$Name

string = [line 142]

The name of the product

phpALUGen_ProductInfo::\$VCode

string = [line 160]

The VCode for the product (Public Key)

phpALUGen_ProductInfo::\$Version

string = [line 148]

The version of the product

Class phpALUGen_ProductLibrary

[line 40]

A class for managing a product library

The base class and functions for manipulating a product library, and performing

searches, adding products, removing products, etc.

- **Package** phpALUGen

boolean function phpALUGen_ProductLibrary::addProduct(\$newProduct) [*line 113*]

Function Parameters:

- [*phpALUGen_ProductInfo*](#) **\$newProduct** The information of the product to add

Adds a product to the library.

array function phpALUGen_ProductLibrary::getAllProducts() [*line 92*]

Retrieves product information for all products in the library.

string function phpALUGen_ProductLibrary::getProductsIni() [*line 159*]

Returns a suitable products.ini file from the products in the library.

function phpALUGen_ProductLibrary::loadProductsIni(\$data, \$remove, \$overwrite) [*line 186*]

Function Parameters:

- *string* **\$data** The products.ini data
- *boolean* **\$remove** Whether or not to remove all existing products first (true=remove all products)
- *boolean* **\$overwrite** Whether or not to overwrite an existing product with the same name and version (true=overwrite)

Loads products from a products.ini-formatted string.

boolean function phpALUGen_ProductLibrary::removeProduct(\$prodName, \$prodVer) [*line 137*]

Function Parameters:

- *string* **\$prodName** Name of the product to search for
- *string* **\$prodVer** Version of the product to search for

Removes a product's information from the library.

Uses name and version to search for a product. If the product exists, it removes the product from the cache and library permanently.

phpALUGen_ProductInfo false function *phpALUGen_ProductLibrary::retrieveProduct(\$prodName, \$prodVer)* [line 65]

Function Parameters:

- *string* **\$prodName** Name of the product to search for
- *string* **\$prodVer** Version of the product to search for

Retrieves a product's information from the library.

Uses name and version to search for a product. If the product has already been loaded, it uses a cached copy if the product's information

Class phpALUGen_ProductLibrary_file

[line 430]

A file-based interface to the Product Library

- **Package** phpALUGen

Constructor function *phpALUGen_ProductLibrary_file::phpALUGen_ProductLibrary_file(\$file_name)* [line 445]

Function Parameters:

- *string* **\$file_name** The name of the file (in products.ini format)

Constructor Function.

Loads the .ini file and parses it

Class phpALUGen_ProductLibrary_mysql

[line 267]

A MySQL interface to the Product Library

- **Package** phpALUGen

Constructor function phpALUGen_ProductLibrary_mysql::phpALUGen_ProductLibrary_mysql(\$db_server, \$db_user, \$db_pass, \$db_name, \$db_table) [line 293]

Function Parameters:

- *string* **\$db_server** Database Server
- *string* **\$db_user** Database Username
- *string* **\$db_pass** Database Password
- *string* **\$db_name** Database Name (the name of the database)
- *string* **\$db_table** Table to be used inside the database

Constructor Function.

Connects to and selects the database, and keeps the connection available

Appendices

Appendix A - Class Trees

Package phpALUGen

phpALUGen

- [phpALUGen](#)

phpALUGen_License

- [phpALUGen License](#)

phpALUGen_ProductInfo

- [phpALUGen_ProductInfo](#)

phpALUGen_ProductLibrary

- [phpALUGen_ProductLibrary](#)
 - [phpALUGen_ProductLibrary_file](#)
 - [phpALUGen_ProductLibrary_mysql](#)

Index

C

constructor phpALUGen ProductLibrary file::phpALUGen ProductLibrary file()	11
<i>Constructor Function.</i>	
constructor phpALUGen ProductLibrary mysql::phpALUGen ProductLibrary mysql()	12
<i>Constructor Function.</i>	

K

keyTools.php	2
<i>The file for key generation and manipulation functions and classes.</i>	

M

miscClasses.php	3
<i>The file for miscellaneous classes and definitions.</i>	

P

phpALUGen.php	4
<i>The main include file for the phpALUGen library.</i>	
productLibrary.php	5
<i>The file for product library functions and classes</i>	
phpALUGen	6
<i>The main phpALUGen class.</i>	
phpALUGen::genKey()	6
<i>A function to generate liberation keys.</i>	
phpALUGen::genProductKeys()	6
<i>Generates new product keys (to be used in a new product) of a given number of bits.</i>	
phpALUGen License	7
<i>A class for handling license information.</i>	
phpALUGen License::\$Expiration	7
<i>The expiration date of the license (YYYY/MM/DD)</i>	
phpALUGen License::\$LicenseClass	7
<i>The class of the license (Single user or Multiple)</i>	
phpALUGen License::\$Licensee	7
<i>The user the license is for</i>	
phpALUGen License::\$LicenseKey	7
<i>The license key (RSA signature)</i>	
phpALUGen License::\$LicenseType	8
<i>The type of the license (Periodic (1), Permanent (2), or Time Locked (3))</i>	
phpALUGen License::\$MaxCount	8
<i>The maximum number of times the product can be used</i>	

phpALUGen_License::\$ProductKey	8
<i>The VCode of the product (Public Key)</i>	
phpALUGen_License::\$ProductName	8
<i>The name of the product the license is for</i>	
phpALUGen_License::\$ProductVer	8
<i>The version of the product the license is for</i>	
phpALUGen_License::\$RegisteredDate	8
<i>The date the license was generated (YYYY/MM/DD)</i>	
phpALUGen_License::save()	8
<i>Generates a copy of this license to use for a liberation key.</i>	
phpALUGen_License::toString()	8
<i>Generates a copy of the license's information in a string.</i>	
phpALUGen_ProductInfo	9
<i>A class for storing product information.</i>	
phpALUGen_ProductInfo::\$GCode	9
<i>The GCode for the product (Private Key)</i>	
phpALUGen_ProductInfo::\$Name	9
<i>The name of the product</i>	
phpALUGen_ProductInfo::\$VCode	9
<i>The VCode for the product (Public Key)</i>	
phpALUGen_ProductInfo::\$Version	9
<i>The version of the product</i>	
phpALUGen_ProductLibrary	9
<i>A class for managing a product library</i>	
phpALUGen_ProductLibrary::addProduct()	10
<i>Adds a product to the library.</i>	
phpALUGen_ProductLibrary::getAllProducts()	10
<i>Retrieves product information for all products in the library.</i>	
phpALUGen_ProductLibrary::getProductsIni()	10
<i>Returns a suitable products.ini file from the products in the library.</i>	
phpALUGen_ProductLibrary::loadProductsIni()	10
<i>Loads products from a products.ini-formatted string.</i>	
phpALUGen_ProductLibrary::removeProduct()	10
<i>Removes a product's information from the library.</i>	
phpALUGen_ProductLibrary::retrieveProduct()	11
<i>Retrieves a product's information from the library.</i>	
phpALUGen_ProductLibrary_file	11
<i>A file-based interface to the Product Library</i>	
phpALUGen_ProductLibrary_mysql	12
<i>A MySQL interface to the Product Library</i>	