

phpALUGen Documentation for phpALUGen Developers



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
Method get32Bit	7
Method getLockAndUserFromInstallCode	7
Method getRSAKeys	7
Method put32Bit	8
Method RSASign	8
Class phpALUGen_License	9
Var \$Expiration	9
Var \$LicenseClass	9
Var \$Licensee	9
Var \$LicenseKey	9
Var \$LicenseType	9
Var \$MaxCount	10
Var \$ProductKey	10
Var \$ProductName	10
Var \$ProductVer	10
Var \$RegisteredDate	10
Method save	10
Method toString	10
Class phpALUGen_ProductInfo	11
Var \$GCode	11
Var \$Name	11
Var \$VCode	11
Var \$Version	11
Class phpALUGen_ProductLibrary	11
Var \$gotWholeListing	12
Var \$products	12
Method addProduct	12
Method getAllProducts	12
Method getProductsIni	12
Method loadProductsIni	13
Method removeProduct	13
Method retrieveProduct	13
Class phpALUGen_ProductLibrary_file	14
Var \$fileName	14

Constructor phpALUGen_ProductLibrary_file	14
Method searchForAllProducts	14
Method searchForProduct	15
Method specialAddProduct	15
Method specialRemoveProduct	15
Class phpALUGen_ProductLibrary_mysql	16
Var \$dbLink	16
Var \$dbTable	16
Constructor phpALUGen_ProductLibrary_mysql	17
Method searchForAllProducts	17
Method searchForProduct	17
Method specialAddProduct	18
Method specialRemoveProduct	18
Appendices	19
Appendix A - Class Trees	20
phpALUGen	20

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.

integer function phpALUGen::get32Bit(\$num) [*line 292*]

Function Parameters:

- *string* **\$num** Input characters

A function to get a 32-bit number from 4 characters.

- **Access** private

function phpALUGen::getLockAndUserFromInstallCode(\$strReq, &\$strLock, &\$strUser) [*line 106*]

Function Parameters:

- *string* **\$strReq** The installation code
- *string* **\$strLock** The Lockcode (pass by reference, will be set by this function)
- *string* **\$strUser** The user (pass by reference, will be set by this function)

Gets the lockcode and user from and installation code.

- **Access** private

function phpALUGen::getRSAKeys(\$VCode, \$GCode, &\$privMod, &\$privExp, &\$pubExp) [*line 130*]

Function Parameters:

- *string* **\$VCode** The VCode to be decoded (contains public key)
- *string* **\$GCode** The GCode to be decoded (contains private key)
- *mixed* **\$privMod** The modulus of the keys
- *mixed* **\$privExp** The exponent of the private key
- *mixed* **\$pubExp** The exponent of the public key

Gets the RSA keys for signing from a VCode and GCode.

The modulus and exponents will be set by the function, and should be passed by reference.

- **Access** private

string function phpALUGen::put32Bit(\$num, \$math_wrapper) [*line 311*]

Function Parameters:

- *integer* **\$num** The number to be converted
- *mixed* **\$math_wrapper** A Crypt/RSA Math Wrapper

A function to create a 4 character string from a 32-bit number.

- **Access** private

string function phpALUGen::RSASign(\$data, \$privMod, \$privExp) [*line 177*]

Function Parameters:

- *string* **\$data** The data to be signed
- *mixed* **\$privMod** The modulus of the key used for signing
- *mixed* **\$privExp** The exponent of the key used for signing

Creates an RSA signature of the data.

- **Access** private

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

phpALUGen_ProductLibrary::\$gotWholeListing

boolean = [line 46]

Whether or not the entire product listing has been retrieved from the library. (Internal Use Only)

- **Access** private

phpALUGen_ProductLibrary::\$products

array = [line 53]

A cache of the products already requested from the library. (Internal Use Only)

- **Access** private

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

phpALUGen_ProductLibrary_file::\$fileName

string = [line 436]

The filename of the library file

- **Access** private

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

array function phpALUGen_ProductLibrary_file::searchForAllProducts() [line 480]

Specialized product-searching routine

Searches the database for all products, and assembles an array of phpALUGen_ProductInfo classes

- **Access** private

phpALUGen_ProductInfo false function *phpALUGen_ProductLibrary_file::searchForProduct*(\$prodName, \$prodVer) [line 466]

Function Parameters:

- *string* **\$prodName** The product name to search for
- *string* **\$prodVer** The version of the product to search for

Specialized product-searching routine

Searches the database for the product, and assembles a *phpALUGen_ProductInfo* class if it is found

- **Access** private

boolean function *phpALUGen_ProductLibrary_file::specialAddProduct*(\$newProduct) [line 493]

Function Parameters:

- [*phpALUGen_ProductInfo*](#) **\$newProduct** Product info to add

Specialized product-adding routine

- **Access** private

boolean function *phpALUGen_ProductLibrary_file::specialRemoveProduct*(\$prodName, \$prodVer) [line 510]

Function Parameters:

- *string* **\$prodName** Name of product to remove
- *string* **\$prodVer** Version of product to remove

Specialized product-removing routine

- **Access** private

Class phpALUGen_ProductLibrary_mysql

[line 267]

A MySQL interface to the Product Library

- **Package** phpALUGen

phpALUGen_ProductLibrary_mysql::\$dbLink

resource = [line 273]

The resource of the link to the database

- **Access** private

phpALUGen_ProductLibrary_mysql::\$dbTable

string = [line 280]

The table the library is in

- **Access** private

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

array function phpALUGen_ProductLibrary_mysql::searchForAllProducts() [line 362]

Specialized product-searching routine

Searches the database for all products, and assembles an array of phpALUGen_ProductInfo classes

- **Access** private

phpALUGen_ProductInfo|false function phpALUGen_ProductLibrary_mysql::searchForProduct(\$prodName, \$prodVer) [line 329]

Function Parameters:

- *string* **\$prodName** The product name to search for
- *string* **\$prodVer** The version of the product to search for

Specialized product-searching routine

Searches the database for the product, and assembles a phpALUGen_ProductInfo class if it is found

- **Access** private

boolean function phpALUGen_ProductLibrary_mysql::specialAddProduct(\$newProduct) [*line 393*]

Function Parameters:

- [*phpALUGen_ProductInfo*](#) **\$newProduct** Product info to add

Specialized product-adding routine

- **Access** private

boolean function phpALUGen_ProductLibrary_mysql::specialRemoveProduct(\$prodName, \$prodVer) [*line 412*]

Function Parameters:

- *string* **\$prodName** Name of product to remove
- *string* **\$prodVer** Version of product to remove

Specialized product-removing routine

- **Access** private

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()	14
<i>Constructor Function.</i>	
constructor phpALUGen ProductLibrary mysql::phpALUGen ProductLibrary mysql()	17
<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::get32Bit()	7
<i>A function to get a 32-bit number from 4 characters.</i>	
phpALUGen::getLockAndUserFromInstallCode()	7
<i>Gets the lockcode and user from and installation code.</i>	
phpALUGen::getRSAKeys()	7
<i>Gets the RSA keys for signing from a VCode and GCode.</i>	
phpALUGen::put32Bit()	8
<i>A function to create a 4 character string from a 32-bit number.</i>	
phpALUGen::RSASign()	8
<i>Creates an RSA signature of the data.</i>	
phpALUGen License	9
<i>A class for handling license information.</i>	
phpALUGen License::\$Expiration	9
<i>The expiration date of the license (YYYY/MM/DD)</i>	

phpALUGen_License::\$LicenseClass	9
<i>The class of the license (Single user or Multiple)</i>	
phpALUGen_License::\$Licensee	9
<i>The user the license is for</i>	
phpALUGen_License::\$LicenseKey	9
<i>The license key (RSA signature)</i>	
phpALUGen_License::\$LicenseType	9
<i>The type of the license (Periodic (1), Permanent (2), or Time Locked (3))</i>	
phpALUGen_License::\$MaxCount	10
<i>The maximum number of times the product can be used</i>	
phpALUGen_License::\$ProductKey	10
<i>The VCode of the product (Public Key)</i>	
phpALUGen_License::\$ProductName	10
<i>The name of the product the license is for</i>	
phpALUGen_License::\$ProductVer	10
<i>The version of the product the license is for</i>	
phpALUGen_License::\$RegisteredDate	10
<i>The date the license was generated (YYYY/MM/DD)</i>	
phpALUGen_License::save()	10
<i>Generates a copy of this license to use for a liberation key.</i>	
phpALUGen_License::toString()	10
<i>Generates a copy of the license's information in a string.</i>	
phpALUGen_ProductInfo	11
<i>A class for storing product information.</i>	
phpALUGen_ProductInfo::\$GCode	11
<i>The GCode for the product (Private Key)</i>	
phpALUGen_ProductInfo::\$Name	11
<i>The name of the product</i>	
phpALUGen_ProductInfo::\$VCode	11
<i>The VCode for the product (Public Key)</i>	
phpALUGen_ProductInfo::\$Version	11
<i>The version of the product</i>	
phpALUGen_ProductLibrary	11
<i>A class for managing a product library</i>	
phpALUGen_ProductLibrary::\$gotWholeListing	12
<i>Whether or not the entire product listing has been retrieved from the library. (Internal Use Only)</i>	
phpALUGen_ProductLibrary::\$products	12
<i>A cache of the products already requested from the library. (Internal Use Only)</i>	
phpALUGen_ProductLibrary::addProduct()	12
<i>Adds a product to the library.</i>	
phpALUGen_ProductLibrary::getAllProducts()	12
<i>Retrieves product information for all products in the library.</i>	
phpALUGen_ProductLibrary::getProductsIni()	12
<i>Returns a suitable products.ini file from the products in the library.</i>	
phpALUGen_ProductLibrary::loadProductsIni()	13
<i>Loads products from a products.ini-formatted string.</i>	
phpALUGen_ProductLibrary::removeProduct()	13
<i>Removes a product's information from the library.</i>	
phpALUGen_ProductLibrary::retrieveProduct()	13
<i>Retrieves a product's information from the library.</i>	
phpALUGen_ProductLibrary_file	14
<i>A file-based interface to the Product Library</i>	
phpALUGen_ProductLibrary_file::\$fileName	14

<i>The filename of the library file</i>	
phpALUGen_ProductLibrary_file::searchForAllProducts()	14
<i>Specialized product-searching routine</i>	
phpALUGen_ProductLibrary_file::searchForProduct()	15
<i>Specialized product-searching routine</i>	
phpALUGen_ProductLibrary_file::specialAddProduct()	15
<i>Specialized product-adding routine</i>	
phpALUGen_ProductLibrary_file::specialRemoveProduct()	15
<i>Specialized product-removing routine</i>	
phpALUGen_ProductLibrary_mysql	16
<i>A MySQL interface to the Product Library</i>	
phpALUGen_ProductLibrary_mysql::\$dbLink	16
<i>The resource of the link to the database</i>	
phpALUGen_ProductLibrary_mysql::\$dbTable	16
<i>The table the library is in</i>	
phpALUGen_ProductLibrary_mysql::searchForAllProducts()	17
<i>Specialized product-searching routine</i>	
phpALUGen_ProductLibrary_mysql::searchForProduct()	17
<i>Specialized product-searching routine</i>	
phpALUGen_ProductLibrary_mysql::specialAddProduct()	18
<i>Specialized product-adding routine</i>	
phpALUGen_ProductLibrary_mysql::specialRemoveProduct()	18
<i>Specialized product-removing routine</i>	