GNU Privacy Guard

Introduction

This module allows you to interact with » gnupg.

Installing/Configuring

Requirements

The gnupg extension requires PHP 4.3. To use this extension in an OO style, PHP 5 is required.

This extension requires the » gpgme library

Installation

The gnupg-extension is not bundled with PHP. It is a » PECL extension and can be located here: » http://pecl.php.net/package/gnupg.

Runtime Configuration

This extension has no configuration directives defined in *php.ini*.

Resource Types

This extension has no resource types defined.

Predefined Constants

GNUPG_SIG_MODE_NORMAL (integer)

GNUPG_SIG_MODE_DETACH (integer)

GNUPG_SIG_MODE_CLEAR (integer)

GNUPG_VALIDITY_UNKNOWN (integer)

GNUPG_VALIDITY_UNDEFINED (integer)

GNUPG_VALIDITY_NEVER (integer)

GNUPG_VALIDITY_MARGINAL (integer)

GNUPG_VALIDITY_FULL (integer)

GNUPG_VALIDITY_ULTIMATE (integer)

GNUPG_PROTOCOL_OpenPGP (integer)

GNUPG_PROTOCOL_CMS (integer)

GNUPG_SIGSUM_VALID (integer)

GNUPG_SIGSUM_GREEN (integer)

GNUPG_SIGSUM_RED (integer)

GNUPG_SIGSUM_KEY_REVOKED (integer)

GNUPG_SIGSUM_KEY_EXPIRED (integer)

GNUPG_SIGSUM_KEY_MISSING (integer)

GNUPG_SIGSUM_SIG_EXPIRED (integer)

GNUPG_SIGSUM_CRL_MISSING (integer)

GNUPG_SIGSUM_CRL_TOO_OLD (integer)

GNUPG_SIGSUM_BAD_POLICY (integer)

GNUPG_SIGSUM_SYS_ERROR (integer)

GNUPG_ERROR_WARNING (integer)

GNUPG_ERROR_EXCEPTION (integer)

GNUPG_ERROR_SILENT (integer)

Examples

Clearsign text

This example will clearsign a given text.

Example #1 - gnupg clearsign example (procedural)

```
<?php
// init gnupg
$res = gnupg_init();
// not really needed. Clearsign is default
gnupg_setsignmode($res,GNUPG_SIG_MODE_CLEAR);
// add key with passphrase 'test' for signing
gnupg_addsignkey($res,"8660281B6051D071D94B5B230549F9DC851566DC","test");
// sign
$signed = gnupg_sign($res,"just a test");
echo $signed;
?>
```

Example #2 - gnupg clearsign example (OO)

```
<?php
// new class
$gnupg = new gnupg();
// not really needed. Clearsign is default
$gnupg->setsignmode(gnupg::SIG_MODE_CLEAR);
// add key with passphrase 'test' for signing
$gnupg->addsignkey("8660281B6051D071D94B5B230549F9DC851566DC","test");
// sign
$signed = $gnupg->sign("just a test");
echo $signed;
?>
```

Example #3 - keylistiterator

This extension also comes with an Iterator for your keyring.

```
<?php
// create a new iterator for listing all public keys that matches 'example'
$iterator = new gnupg_keylistiterator("example");
foreach($iterator as $fingerprint => $userid){
   echo $fingerprint." -> ".$userid."\n";
}
?>
```

GnuPG Functions

Notes

This extension makes use of the keyring of the current user. This keyring is normally located in ~./.gnupg/. To specify a custom location, store the path to the keyring in the environment variable GNUPGHOME. See putenv for more information how to do this.

Some functions require the specification of a key. This specification can be anything that refers to an unique key (userid, key-id, fingerprint, ...). This documentation uses the fingerprint in all examples.

gnupg_adddecryptkey

gnupg_adddecryptkey -- Add a key for decryption

Description

bool **gnupg_adddecryptkey** (resource `$identifier`, string `$fingerprint`, string `$passphrase`)

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

`fingerprint`
The fingerprint key.

`passphrase`
The pass phrase.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #4 - Procedural gnupg_adddecryptkey() example

```
<?php
$res = gnupg_init();
gnupg_adddecryptkey($res,"8660281B6051D071D94B5B230549F9DC851566DC","test");
?>
```

Example #5 - OO gnupg_adddecryptkey() example

```
<?php
$gpg = new gnupg();
$gpg -> adddecryptkey("8660281B6051D071D94B5B230549F9DC851566DC","test");
?>
```

gnupg_addencryptkey

gnupg_addencryptkey -- Add a key for encryption

Description

bool **gnupg_addencryptkey** (resource `$identifier`, string `$fingerprint`)

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

`fingerprint`
The fingerprint key.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #6 - Procedural gnupg_addencryptkey() example

```
<?php
$res = gnupg_init();
gnupg_addencryptkey($res,"8660281B6051D071D94B5B230549F9DC851566DC");
?>
```

Example #7 - OO gnupg_addencryptkey() example

```
<?php
$gpg = new gnupg();
$gpg -> addencryptkey("8660281B6051D071D94B5B230549F9DC851566DC");
?>
```

gnupg_addsignkey

gnupg_addsignkey -- Add a key for signing

Description

bool **gnupg_addsignkey** (resource `$identifier`, string `$fingerprint` [, string `$passphrase`])

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

`fingerprint`
The fingerprint key.

`passphrase`
The pass phrase.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #8 - Procedural gnupg_addsignkey() example

```
<?php
$res = gnupg_init();
gnupg_addsignkey($res,"8660281B6051D071D94B5B230549F9DC851566DC","test");
?>
```

Example #9 - OO gnupg_addsignkey() example

```
<?php
$gpg = new gnupg();
$gpg -> addsignkey("8660281B6051D071D94B5B230549F9DC851566DC","test");
?>
```

gnupg_cleardecryptkeys

gnupg_cleardecryptkeys -- Removes all keys which were set for decryption before

Description

bool **gnupg_cleardecryptkeys** (resource `$identifier`)

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #10 - Procedural gnupg_cleardecryptkeys() example

```
<?php
$res = gnupg_init();
gnupg_cleardecryptkeys($res);
?>
```

Example #11 - OO gnupg_cleardecryptkeys() example

```
<?php
$gpg = new gnupg();
$gpg -> cleardecryptkeys();
?>
```

gnupg_clearencryptkeys

gnupg_clearencryptkeys -- Removes all keys which were set for encryption before

Description

bool **gnupg_clearencryptkeys** (resource `$identifier`)

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #12 - Procedural gnupg_clearencryptkeys() example

```
<?php
$res = gnupg_init();
gnupg_clearencryptkeys($res);
?>
```

Example #13 - OO gnupg_clearencryptkeys() example

```
<?php
$gpg = new gnupg();
$gpg -> clearencryptkeys();
?>
```

gnupg_clearsignkeys

gnupg_clearsignkeys -- Removes all keys which were set for signing before

Description

bool **gnupg_clearsignkeys** (resource `$identifier`)

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #14 - Procedural gnupg_clearsignkeys() example

```
<?php
$res = gnupg_init();
gnupg_clearsignkeys($res);
?>
```

Example #15 - OO gnupg_clearsignkeys() example

```
<?php
$gpg = new gnupg();
$gpg -> clearsignkeys();
?>
```

gnupg_decrypt

gnupg_decrypt -- Decrypts a given text

Description

string **gnupg_decrypt** (resource `$identifier`, string `$text`)

Decrypts the given text with the keys, which were set with gnupg_adddecryptkey before.

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

`text`
The text being decrypted.

Return Values

On success, this function returns the decrypted text. On failure, this function returns **FALSE**.

Examples

Example #16 - Procedural gnupg_decrypt() example

```
<?php
$res = gnupg_init();
gnupg_adddecryptkey($res,"8660281B6051D071D94B5B230549F9DC851566DC","test");
$plain = gnupg_decrypt($res,$encrypted_text);
echo $plain;
?>
```

Example #17 - OO gnupg_encrypt() example

```
<?php
$gpg = new gnupg();
$gpg -> adddecryptkey("8660281B6051D071D94B5B230549F9DC851566DC","test");
$plain = $gpg -> decrypt($encrypted_text);
echo $plain;
?>
```

gnupg_decryptverify

gnupg_decryptverify -- Decrypts and verifies a given text

Description

array **gnupg_decryptverify** (resource `$identifier`, string `$text`, string `&$plaintext`)

Decrypts and verifies a given text and returns information about the signature.

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

`text`
The text being decrypted.

`plaintext`
The parameter *`plaintext`* gets filled with the decrypted text.

Return Values

On success, this function returns information about the signature and fills the *`plaintext`* parameter with the decrypted text. On failure, this function returns **FALSE**.

Examples

Example #18 - Procedural gnupg_decryptverify() example

```
<?php
$plaintext = "";
$res = gnupg_init();
gnupg_adddecryptkey($res,"8660281B6051D071D94B5B230549F9DC851566DC","test");
$info = gnupg_decryptverify($res,$text,$plaintext);
print_r($info);
?>
```

Example #19 - OO gnupg_decryptverify() example

```
<?php
$plaintext = "";
$gpg = new gnupg();
$gpg -> adddecryptkey("8660281B6051D071D94B5B230549F9DC851566DC","test");
```

```
$info = $gpg -> decryptverify($text,$plaintext);
print_r($info);
?>
```

gnupg_encrypt

gnupg_encrypt -- Encrypts a given text

Description

string **gnupg_encrypt** (resource `$identifier`, string `$plaintext`)

Encrypts the given *`plaintext`* with the keys, which were set with gnupg_addencryptkey before and returns the encrypted text.

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

`plaintext`
The text being encrypted.

Return Values

On success, this function returns the encrypted text. On failure, this function returns **FALSE**.

Examples

Example #20 - Procedural gnupg_encrypt() example

```
<?php
$res = gnupg_init();
gnupg_addencryptkey($res,"8660281B6051D071D94B5B230549F9DC851566DC");
$enc = gnupg_encrypt($res, "just a test");
echo $enc;
?>
```

Example #21 - OO gnupg_encrypt() example

```
<?php
$gpg = new gnupg();
$gpg -> addencryptkey("8660281B6051D071D94B5B230549F9DC851566DC");
$enc = $gpg -> encrypt("just a test");
echo $enc;
?>
```

gnupg_encryptsign

gnupg_encryptsign -- Encrypts and signs a given text

Description

string **gnupg_encryptsign** (resource `$identifier`, string `$plaintext`)

Encrypts and signs the given *`plaintext`* with the keys, which were set with gnupg_addsignkey and gnupg_addencryptkey before and returns the encrypted and signed text.

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

`plaintext`
The text being encrypted.

Return Values

On success, this function returns the encrypted and signed text. On failure, this function returns **FALSE**.

Examples

Example #22 - Procedural gnupg_encryptsign() example

```
<?php
$res = gnupg_init();
gnupg_addencryptkey($res,"8660281B6051D071D94B5B230549F9DC851566DC");
gnupg_addsignkey($res,"8660281B6051D071D94B5B230549F9DC851566DC","test");
$enc = gnupg_encryptsign($res, "just a test");
echo $enc;
?>
```

Example #23 - OO gnupg_encryptsign() example

```
<?php
$gpg = new gnupg();
$gpg -> addencryptkey("8660281B6051D071D94B5B230549F9DC851566DC");
$gpg -> addsignkey("8660281B6051D071D94B5B230549F9DC851566DC","test");
$enc = $gpg -> encryptsign("just a test");
```

```
echo $enc;
?>
```

gnupg_export

gnupg_export -- Exports a key

Description

string **gnupg_export** (resource `$identifier`, string `$fingerprint`)

Exports the key *`fingerprint`*.

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

`fingerprint`
The fingerprint key.

Return Values

On success, this function returns the keydata. On failure, this function returns **FALSE**.

Examples

Example #24 - Procedural gnupg_export() example

```
<?php
$res = gnupg_init();
$export = gnupg_export($res,"8660281B6051D071D94B5B230549F9DC851566DC");
echo $export;
?>
```

Example #25 - OO gnupg_export() example

```
<?php
$gpg = new gnupg();
$export = $gpg -> export("8660281B6051D071D94B5B230549F9DC851566DC");
?>
```

gnupg_geterror

gnupg_geterror -- Returns the errortext, if a function fails

Description

string **gnupg_geterror** (resource `$identifier`)

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

Return Values

Returns an errortext, if an error has occurred, otherwise **FALSE**.

Examples

Example #26 - Procedural gnupg_geterror() example

```
<?php
$res = gnupg_init();
echo gnupg_geterror($res);
?>
```

Example #27 - OO gnupg_geterror() example

```
<?php
$gpg = new gnupg();
echo $gpg -> geterror();
?>
```

gnupg_getprotocol

gnupg_getprotocol -- Returns the currently active protocol for all operations

Description

int **gnupg_getprotocol** (resource `$identifier`)

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

Return Values

Returns the currently active protocol, which can be one of **GNUPG_PROTOCOL_OpenPGP** or **GNUPG_PROTOCOL_CMS**.

Examples

Example #28 - Procedural gnupg_getprotocol() example

```
<?php
$res = gnupg_init();
echo gnupg_getprotocol($res);
?>
```

Example #29 - OO gnupg_getprotocol() example

```
<?php
$gpg = new gnupg();
echo $gpg -> getprotocol();
?>
```

gnupg_import

gnupg_import -- Imports a key

Description

array **gnupg_import** (resource `$identifier`, string `$keydata`)

Imports the key *`keydata`* and returns an array with information about the importprocess.

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

`keydata`
The data key that is being imported.

Return Values

On success, this function returns and info-array about the importprocess. On failure, this function returns **FALSE**.

Examples

Example #30 - Procedural gnupg_import() example

```
<?php
$res = gnupg_init();
$info = gnupg_import($res,$keydata);
print_r($info);
?>
```

Example #31 - OO gnupg_import() example

```
<?php
$gpg = new gnupg();
$info = $gpg -> import($keydata);
print_r($info);
?>
```

gnupg_init

gnupg_init -- Initialize a connection

Description

resource **gnupg_init** (void)

Parameters

This function has no parameters.

Return Values

A GnuPG resource connection used by other GnuPG functions.

Examples

Example #32 - Procedural gnupg_init() example

```
<?php
$res = gnupg_init();
?>
```

Example #33 - OO gnupg initializer example

```
<?php
$gpg = new gnupg();
?>
```

gnupg_keyinfo

gnupg_keyinfo -- Returns an array with information about all keys that matches the given pattern

Description

array **gnupg_keyinfo** (resource `$identifier`, string `$pattern`)

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

`pattern`
The pattern being checked against the keys.

Return Values

Returns an array with information about all keys that matches the given pattern or **FALSE**, if an error has occurred.

Examples

Example #34 - Procedural gnupg_keyinfo() example

```
<?php
$res = gnupg_init();
$info = gnupg_keyinfo($res, 'test');
print_r($info);
?>
```

Example #35 - OO gnupg_keyinfo() example

```
<?php
$gpg = new gnupg();
$info = $gpg -> keyinfo("test");
print_r($info);
?>
```

gnupg_setarmor

gnupg_setarmor -- Toggle armored output

Description

bool **gnupg_setarmor** (resource `$identifier`, int `$armor`)

Toggle the armored output.

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

`armor`
Pass a non-zero integer-value to this function to enable armored-output (default). Pass 0 to disable armored output.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #36 - Procedural gnupg_setarmor() example

```
<?php
$res = gnupg_init();
gnupg_setarmor($res,1); // enable armored output;
gnupg_setarmor($res,0); // disable armored output;
?>
```

Example #37 - OO gnupg_setarmor() example

```
<?php
$gpg = new gnupg();
$gpg -> setarmor(1); // enable armored output;
$gpg -> setarmor(0); // disable armored output;
?>
```

gnupg_seterrormode

gnupg_seterrormode -- Sets the mode for error_reporting

Description

void **gnupg_seterrormode** (resource `$identifier`, int `$errormode`)

Sets the mode for error_reporting.

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

`errormode`
The error mode. *`errormode`* takes a constant indicating what type of error_reporting should be used. The possible values are **GNUPG_ERROR_WARNING**, **GNUPG_ERROR_EXCEPTION** and **GNUPG_ERROR_SILENT**. By default **GNUPG_ERROR_SILENT** is used.

Return Values

No value is returned.

Examples

Example #38 - Procedural gnupg_seterrormode() example

```
<?php
$res = gnupg_init();
gnupg_seterrormode($res,GNUPG_ERROR_WARNING); // raise a PHP-Warning in case
of an error
?>
```

Example #39 - OO gnupg_seterrormode() example

```
<?php
$gpg = new gnupg();
$gpg -> seterrormode(gnupg::ERROR_EXCEPTION); // throw an exception in case
of an error
?>
```

gnupg_setsignmode

gnupg_setsignmode -- Sets the mode for signing

Description

bool **gnupg_setsignmode** (resource `$identifier`, int `$signmode`)

Sets the mode for signing.

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

`sigmode`
The mode for signing. *`signmode`* takes a constant indicating what type of signature should be produced. The possible values are **GNUPG_SIG_MODE_NORMAL**, **GNUPG_SIG_MODE_DETACH** and **GNUPG_SIG_MODE_CLEAR**. By default **GNUPG_SIG_MODE_CLEAR** is used.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #40 - Procedural gnupg_setsignmode() example

```
<?php
$res = gnupg_init();
gnupg_setsignmode($res,GNUPG_SIG_MODE_DETACH); // produce a detached
signature
?>
```

Example #41 - OO gnupg_setsignmode() example

```
<?php
$gpg = new gnupg();
$gpg -> setsignmode(gnupg::SIG_MODE_DETACH); // produce a detached signature
?>
```

gnupg_sign

gnupg_sign -- Signs a given text

Description

string **gnupg_sign** (resource `$identifier`, string `$plaintext`)

Signs the given *`plaintext`* with the keys, which were set with gnupg_addsignkey before and returns the signed text or the signature, depending on what was set with gnupg_setsignmode.

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

`plaintext`
The plain text being signed.

Return Values

On success, this function returns the signed text or the signature. On failure, this function returns **FALSE**.

Examples

Example #42 - Procedural gnupg_sign() example

```
<?php
$res = gnupg_init();
gnupg_addsignkey($res,"8660281B6051D071D94B5B230549F9DC851566DC","test");
$signed = gnupg_sign($res, "just a test");
echo $signed;
?>
```

Example #43 - OO gnupg_sign() example

```
<?php
$gpg = new gnupg();
$gpg -> setsignkey("8660281B6051D071D94B5B230549F9DC851566DC","test");
$signed = $gpg -> sign("just a test");
echo $signed;
?>
```

gnupg_verify

gnupg_verify -- Verifies a signed text

Description

array **gnupg_verify** (resource `$identifier`, string `$signed_text`, string `$signature` [, string `&$plaintext`])

Verifies the given *`signed_text`* and returns information about the signature.

Parameters

`identifier`
The gnupg identifier, from a call to gnupg_init() or gnupg.

`signed_text`
The signed text.

`signature`
The signature. To verify a clearsigned text, set signature to **FALSE**.

`plaintext`
The plain text. If this optional parameter is passed, it is filled with the plain text.

Return Values

On success, this function returns informations about the signature. On failure, this function returns **FALSE**.

Examples

Example #44 - Procedural gnupg_verify() example

```
<?php
$plaintext = "";
$res = gnupg_init();
// clearsigned
$info = gnupg_verify($res,$signed_text,false,$plaintext);
print_r($info);
// detached signature
$info = gnupg_verify($res,$signed_text,$signature);
print_r($info);
?>
```

Example #45 - OO gnupg_verify() example

```
<?php
$plaintext = "";
$gpg = new gnupg();
// clearsigned
$info = $gpg -> verify($signed_text,false,$plaintext);
print_r($info);
// detached signature
$info = $gpg -> verify($signed_text,$signature);
print_r($info);
?>
```