

# **DOCUMENTAZIONE TECNICA CONNECTOR PHP**

## **VERSIONE 2.0**

G. Franchitto  
1.0.0  
30/04/2015





# Namespace Index

## Namespace List

Here is a list of all namespaces with brief descriptions:

<b>AIPSCollector</b>	.....	8
<b>PHPMailer</b>	.....	13

# Hierarchical Index

## Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

ERP_TPD_LISTINI .....	15
ERP_TPD_LISTINI_BASE .....	16
ERP_TPD_PRODOTTI.....	17
ERP_TSD_ANAGRAFICHE.....	22
ERP_TSD_SCONTI_SCAGLIONE .....	27
ERP_TSD_UTENTI.....	28
Exception	
phpmailerException.....	64
MySQL.....	30
PHPMailer .....	44
PS_specific_price .....	65
VisionConnectorConfig.....	94
PSObject.....	78
PSCategory.....	66
PSCustomer .....	70
PSFeature .....	73
PSListini.....	75
PSOrder .....	82
PSPayment .....	83
PSProduct.....	84
PSScadenze .....	93
VisionConnectorConfig	
VisionEcommerceConnector.....	96

# Data Structure Index

## Data Structures

Here are the data structures with brief descriptions:

<b>ERP_TPD_LISTINI</b>	15
<b>ERP_TPD_LISTINI_BASE</b>	16
<b>ERP_TPD_PRODOTTI</b>	17
<b>ERP_TSD_ANAGRAFICHE</b>	22
<b>ERP_TSD_SCONTI_SCAGLIONE</b>	27
<b>ERP_TSD_UTENTI</b>	28
<b>MySQL</b>	30
<b>PHPMailer</b>	44
<b>phpmailerException</b>	64
<b>PS_specific_price</b>	65
<b>PSCategory</b>	66
<b>PSCustomer</b>	70
<b>PSFeature</b>	73
<b>PSListini</b>	75
<b>PSObject</b>	78
<b>PSOrder</b>	82
<b>PSPayment</b>	83
<b>PSProduct</b>	84
<b>PSScadenze</b>	93
<b>VisionConnectorConfig</b>	94
<b>VisionEcommerceConnector</b>	96

# File Index

## File List

Here is a list of all files with brief descriptions:

<b>cleanPSCache.php</b>	109
<b>endpnt_export.php</b>	111
<b>endpnt_import.php</b>	112
<b>erp2ps.php</b>	113
<b>ps2erp.php</b>	<b>Error! Bookmark not defined.</b>
<b>classes/PSCategory.php</b>	100
<b>classes/PSCustomer.php</b>	101
<b>classes/PSFeature.php</b>	102
<b>classes/PSListini.php</b>	103
<b>classes/PSObject.php</b>	104
<b>classes/PSOrder.php</b>	105
<b>classes/PSPayment.php</b>	106
<b>classes/PSProduct.php</b>	107
<b>classes/PSScadenze.php</b>	108
<b>conf/config.connector.php</b>	110
<b>lib/class.phpmailer.php</b>	114
<b>lib/ERP.class.php</b>	115
<b>lib/mysql.class.php</b>	116

# Namespace Documentation

## AIPSCorridor Namespace Reference

### Data Structures

- class **MySQL**
- class **PSCategory**
- class **PSCustomer**
- class **PSFeature**
- class **PSListini**
- class **PSObject**
- class **PSOrder**
- class **PSPayment**
- class **PSProduct**
- class **PSScadenze**
- class **VisionConnectorConfig**

### Variables

- const **FILE\_CATEGORY** 'TPD\_PRODOTTI\_CATEGORIE'
- const **FILE\_CATEGORY\_LANG** 'TPD\_PRODOTTI\_LINGUA\_CATEGORIE'
- const **FOLDER\_CATEGORY\_IMG** **PATH\_ABSOLUTE\_AICORRIDOR** . "/img/c/"
- const **TOPMENU\_ENABLECAT** "1"
- const **FILE\_CUSTOMER** "TSD\_UTENTI"
- const **FILE\_ANAGRAFICHE** "TSD\_ANAGRAFICHE"
- const **URL\_WGET\_RELOAD\_ATTRIBUTE**  
"http://devps.all-init.it/modules/blocklayered/blocklayered-attribute-indexer.php?token=f550d3b96e"
- const **URL\_WGET\_RELOAD\_ATTRIBUTE\_URL**  
"http://devps.all-init.it/modules/blocklayered/blocklayered-url-indexer.php?token=f550d3b96e&truncate=1"
- const **FILE\_PRODUCT** "TPD\_PRODOTTI"
- const **FILE\_LISTINI\_BASE** "TPD\_LISTINI\_BASE"
- const **FILE\_LISTINI** "TPD\_LISTINI"
- const **FILE\_SCONTI\_QT** "TSD\_SCONTI\_SCAGLIONE"
- const **LIST\_BASE\_NUMBER** 2
- const **PROPAPAGAZIONE\_LISTINI\_CLIENTI** 0
- **\$FILE\_LOG\_NAME** = "log\_import\_" . date("YmdHis") . ".txt"
- const **FILE\_VERSION\_ERP** "CONN\_VER"
- const **FILE\_INFO** "TPD\_INFO"
- const **FOLDER\_PDF** **PATH\_ABSOLUTE\_AICORRIDOR** . "/img/p/pdf/"
- const **FOLDER\_TMP\_PDF** **PATH\_ABSOLUTE\_AICORRIDOR** . "/img/pdftmp/"
- const **FILE\_PRODUCT\_LINGUA** "TPD\_PRODOTTI\_LINGUA"
- const **ENABLE\_TAGLIE** 0
- const **DELIMITATORE\_TAGLIE** "èéà"
- const **FILL\_PARENT\_CATEGORY** 1
- const **FOLDER\_ATTACHMENT** **PATH\_ABSOLUTE\_AICORRIDOR** . "/download"
- const **DB\_HOST** "localhost"
- const **DB\_USER** 'gfranchitto'
- const **DB\_PASS** 'crekepispì'
- const **DB\_NAME** "devps"
- const **PATH\_ABSOLUTE\_AICORRIDOR** "/home/gfranchitto/domains/devps.all-init.it/public\_html"
- const **FOLDER\_UNZIP** **PATH\_ABSOLUTE\_AICORRIDOR** . "/visionImport/tmp\_data/"
- const **FOLDER\_PS\_IMAGE** **PATH\_ABSOLUTE\_AICORRIDOR** . "/img/p/"



- `const FOLDER_TMP_IMG_PATH_ABSOLUTE_AICONNECTOR . "/img/tmp/"`
- `const FOLDER_IMPORT_ROOT_PATH_ABSOLUTE_AICONNECTOR . "/visionImport/"`
- `const FILE_ERP_DATA "d_oi_b2b.zip"`
- `const FILE_ERP_IMG "oi_b2b_files.zip"`
- `const WIN_CONNECTOR_VER "04"`
- `const SALT_PASSWORD "3vLbIAfx0jM8WBkFuaMRmtwkgdJktPgHFU9G0vBka783cOtiFMb10KW"`
- `const ADMIN_FOLDER "admin4675"`
- `const _PS_ADMIN_DIR_PATH_ABSOLUTE_AICONNECTOR . '/' . ADMIN_FOLDER`
- `const EMAIL_LOG_TO "gfranchitto@all-init.it"`
- `const EMAIL_LOG_TO_2 "gfranchitto2@all-init.it"`
- `const EMAIL_LOG_FROM "notifiche@all-init.it"`
- `const EMAIL_LOG_SUBJECT "[DEVPS - REPORT] Export Product to Ecommerce"`
- `const LANG_NUMBER 1`
- `const SHOP_NAME 'DEVPS Ecommerce'`
- `const SHOP_URL "http://devps.all-init.it/"`
- `const SHOP_LOGO "http://devps.all-init.it/img/logo.jpg"`
- `const SUBJ_ACTIVATEEMAIL "[DEVPS ALLINIT] - Attivazione account"`
- `const FROM_ACTIVATEEMAIL "devps@all-init.it"`

## Detailed Description

Ultimate MySQL Wrapper Class

### Version:

3.0

### Author:

Jeff L. Williams Contributions from Frank P. Walentynowicz Larry Wakeman Nicola Abbiuso Douglas Gintz  
Emre Erkan Vincent van Daal Michael Ansaldi Ryan Tse

## Variable Documentation

`$FILE_LOG_NAME = "log_import_" . date("YmdHis") . ".txt"`

`const _PS_ADMIN_DIR_PATH_ABSOLUTE_AICONNECTOR . '/' . ADMIN_FOLDER`

`const ADMIN_FOLDER "admin4675"`

`const DB_HOST "localhost"`

`const DB_NAME "devps"`

`const DB_PASS 'crekepispi'`

`const DB_USER 'gfranchitto'`

`const DELIMITATORE_TAGLIE "èèà"`

Stringa che delimita le taglie nel codice del prodotto

**const EMAIL\_LOG\_FROM "notifiche@all-init.it"**

**const EMAIL\_LOG\_SUBJECT "[DEVPS - REPORT] Export Product to Ecommerce"**

**const EMAIL\_LOG\_TO "gfranchitto@all-init.it"**

**const EMAIL\_LOG\_TO\_2 "gfranchitto2@all-init.it"**

**const ENABLE\_TAGLIE 0**

Variabile che abilita o meno la gestione delle taglie

**const FILE\_ANAGRAFICHE "TSD\_ANAGRAFICHE"**

**const FILE\_CATEGORY 'TPD\_PRODOTTI\_CATEGORIE'**

Nome del file del tracciato di ERP per le Catogorie

**const FILE\_CATEGORY\_LANG 'TPD\_PRODOTTI\_LINGUA\_CATEGORIE'**

Nome del file del tracciato lingue di ERP per le Categorie

**const FILE\_CUSTOMER "TSD\_UTENTI"**

**const FILE\_ERP\_DATA "d\_oi\_b2b.zip"**

**const FILE\_ERP\_IMG "oi\_b2b\_files.zip"**

**const FILE\_INFO "TPD\_INFO"**

File INFO ecommerce

**const FILE\_LISTINI "TPD\_LISTINI"**

**const FILE\_LISTINI\_BASE "TPD\_LISTINI\_BASE"**

**const FILE\_PRODUCT "TPD\_PRODOTTI"**

Nome file tracciato contenente le informazioni del prodotto

**const FILE\_PRODUCT\_LINGUA "TPD\_PRODOTTI\_LINGUA"**

Nome file tracciato contenente le traduzioni del prodotto

**const FILE\_SCONTI\_QT "TSD\_SCONTI\_SCAGLIONE"**

**const FILE\_VERSION\_ERP "CONN\_VER"**

File contenente le informazioni sulla versione del connector windows usata

**const FILL\_PARENT\_CATEGORY 1**

Flag che abilita l'opzione di associare il prodotto a tutte le sottocategorie della categoria principale padre

**const FOLDER\_ATTACHMENT\_PATH\_ABSOLUTE\_AICONNECTOR . "/download"**

Percorso in cui vengono caricati gli allegati ai prodotti

**const FOLDER\_CATEGORY\_IMG\_PATH\_ABSOLUTE\_AICONNECTOR . "/img/c/"**

Cartella contenente le immagini delle categorie di Prestashop

**const FOLDER\_IMPORT\_ROOT\_PATH\_ABSOLUTE\_AICONNECTOR . "/visionImport/"**

**const FOLDER\_PDF\_PATH\_ABSOLUTE\_AICONNECTOR . "/img/p/pdf/"**

**const FOLDER\_PS\_IMAGE\_PATH\_ABSOLUTE\_AICONNECTOR . "/img/p/"**

**const FOLDER\_TMP\_IMG\_PATH\_ABSOLUTE\_AICONNECTOR . "/img/tmp/"**

**const FOLDER\_TMP\_PDF\_PATH\_ABSOLUTE\_AICONNECTOR . "/img/pdf/tmp/"**

**const FOLDER\_UNZIP\_PATH\_ABSOLUTE\_AICONNECTOR . "/visionImport/tmp\_data/"**

**const FROM\_ACTIVATEEMAIL "devps@all-init.it"**

**const LANG\_NUMBER 1**

**const LIST\_BASE\_NUMBER 2**

**const PATH\_ABSOLUTE\_AICONNECTOR "/home/gfranchitto/domains/devps.all-init.it/public\_html"**

**const PROPAGAZIONE\_LISTINI\_CLIENTI 0**

**const SALT\_PASSWORD "3vLbIAfx0jM8WBkFuaMRmtwkgdJktPgHFU9G0vBka783cOtiFMb10KW"**

**const SHOP\_LOGO "http://devps.all-init.it/img/logo.jpg"**

**const SHOP\_NAME 'DEVPS Ecommerce'**

**const SHOP\_URL "http://devps.all-init.it/"**

**const SUBJ\_ACTIVATEEMAIL "[DEVPS ALLINIT] - Attivazione account"**

**const TOPMENU\_ENABLECAT "1"**

Variabile indicata per abilitare o meno la popolazione automatica del menù TOP

```
const URL_WGET_RELOAD_ATTRIBUTE
```

```
"http://devps.all-init.it/modules/blocklayered/blocklayered-attribute-indexer.php?token=f550d3b96e"
```

```
const URL_WGET_RELOAD_ATTRIBUTE_URL
```

```
"http://devps.all-init.it/modules/blocklayered/blocklayered-url-indexer.php?token=f550d3b96e&truncate=1"
```

```
const WIN_CONNECTOR_VER "04"
```

# PHPMailer Namespace Reference

---

## Detailed Description

**PHPMailer** - PHP email creation and transport class NOTE: Requires PHP version 5 or later

### Author:

Andy Prevost  
Marcus Bointon  
Jim Jagielski

### Copyright:

2010 - 2012 Jim Jagielski  
2004 - 2009 Andy Prevost <http://www.gnu.org/copyleft/lesser.html> GNU Lesser General Public License

PHP email creation and transport class

Exception handler for **PHPMailer**

## PSCategory Namespace Reference

---

### Detailed Description

Gestione Categorie di Prestashop

Classe per la gestione delle categorie di Prestashop Compatibile solo con versioni di prestashop 1.5.x

**Author:**

AlliniT Srls

**Version:**

2.0.0

**Copyright:**

# Data Structure Documentation

## ERP\_TPD\_LISTINI Class Reference

### Data Fields

- **`$_0_Idlistino`**
- **`$_1_Cdprodotto`**
- **`$_2_Cdanag`**
- **`$_3_Cdcategoria`**
- **`$_4_CDcatCliente`**
- **`$_5_Nrprezzo`**
- **`$_6_Dssconto`**
- **`$_7_Dtinizio`**
- **`$_8_Dtfine`**

---

### Field Documentation

**`$_0_Idlistino`**

**`$_1_Cdprodotto`**

**`$_2_Cdanag`**

**`$_3_Cdcategoria`**

**`$_4_CDcatCliente`**

**`$_5_Nrprezzo`**

**`$_6_Dssconto`**

**`$_7_Dtinizio`**

**`$_8_Dtfine`**

---

The documentation for this class was generated from the following file:

- `lib/ERP.class.php`

## ERP\_TPD\_LISTINI\_BASE Class Reference

### Data Fields

- `$_0_Idlistino`
  - `$_1_Cdprodotto`
  - `$_2_CDvaluta`
  - `$_3_NRprezzo`
  - `$_4_DSsconto`
  - `$_5_CDlistino`
- 

### Field Documentation

`$_0_Idlistino`

`$_1_Cdprodotto`

`$_2_CDvaluta`

`$_3_NRprezzo`

`$_4_DSsconto`

`$_5_CDlistino`

---

The documentation for this class was generated from the following file:

- `lib/ERP.class.php`



## ERP\_TPD\_PRODOTTI Class Reference

### Data Fields

- **\$\_0\_Cdprodotto**
- **\$\_1\_Cdmenu**
- **\$\_2\_DSnomeProdotto**
- **\$\_3\_Cdmarca**
- **\$\_4\_Mmdescrizione**
- **\$\_5\_Dskeywords**
- **\$\_6\_Dsdescription**
- **\$\_7\_FLSchedaTecnica**
- **\$\_8\_DSschedaTecnica**
- **\$\_9\_FLimgDettaglio**
- **\$\_10\_DSimgDettaglio**
- **\$\_11\_FlingGrande**
- **\$\_12\_DSimgGrande**
- **\$\_13\_FLimgPiccola**
- **\$\_14\_DSimgPiccola**
- **\$\_15\_FLimgAggiuntiva1**
- **\$\_16\_FLimgAggiuntiva2**
- **\$\_17\_FLimgAggiuntiva3**
- **\$\_18\_FLimgAggiuntiva4**
- **\$\_19\_FLimgAggiuntiva5**
- **\$\_20\_NRprezzoListino**
- **\$\_21\_NRprezzoScontato**
- **\$\_22\_Nrpeso**
- **\$\_23\_NRordineVis**
- **\$\_24\_CdstatoProdotto**
- **\$\_25\_Fgnovita**
- **\$\_26\_Fgdisponibile**
- **\$\_27\_Fgofferta**
- **\$\_28\_Fgvetrina**
- **\$\_29\_Fgfiniserie**
- **\$\_30\_Fgomaggio**
- **\$\_31\_FGrichiastaPreventivo**
- **\$\_32\_DSurlRewrite**
- **\$\_33\_DSurlYoutube**
- **\$\_34\_Dtins**
- **\$\_35\_Cdute\_ins**
- **\$\_36\_Dtmod**
- **\$\_37\_Cdute\_mod**
- **\$\_38\_CDuteDel**
- **\$\_39\_Dtdel**
- **\$\_40\_Fgdel**
- **\$\_41\_TTlastUpdate**
- **\$\_42\_Dssconto**
- **\$\_43\_Nrqta**
- **\$\_44\_Cdiva**
- **\$\_45\_NRqtaConfezione**
- **\$\_46\_NRqtaMinOrdinabile**
- **\$\_47\_DSunitaMisura**
- **\$\_48\_NRqtaMultiplo**

- **\$\_49\_CDcategoria**
-

## **Field Documentation**

**\$\_0\_Cdprodotto**

**\$\_10\_DSimgDettaglio**

**\$\_11\_FlimgGrande**

**\$\_12\_DSimgGrande**

**\$\_13\_FLimgPiccola**

**\$\_14\_DSimgPiccola**

**\$\_15\_FLimgAggiuntiva1**

**\$\_16\_FLimgAggiuntiva2**

**\$\_17\_FLimgAggiuntiva3**

**\$\_18\_FLimgAggiuntiva4**

**\$\_19\_FLimgAggiuntiva5**

**\$\_1\_Cdmenu**

**\$\_20\_NRprezzoListino**

**\$\_21\_NRprezzoScontato**

**\$\_22\_Nrpeso**

**\$\_23\_NRordineVis**

**\$\_24\_CdstatoProdotto**

**\$\_25\_Fgnovita**

**\$\_26\_Fgdisponibile**

**\$\_27\_Fgofferta**

**\$\_28\_Fgvetrina**

**\$\_29\_Fgfineserie**

**\$\_2\_DSnomeProdotto**

**\$\_30\_Fgomaggio**

**\$\_31\_FGrichietaPreventivo**

**\$\_32\_DSurlRewrite**

**\$\_33\_DSurlYoutube**

**\$\_34\_Dtins**

**\$\_35\_Cdute\_ins**

**\$\_36\_Dtmod**

**\$\_37\_Cdute\_mod**

**\$\_38\_CDuteDel**

**\$\_39\_Dtdel**

**\$\_3\_Cdmarca**

**\$\_40\_Fgdel**

**\$\_41\_TTlastUpdate**

**\$\_42\_Dssconto**

**\$\_43\_Nrqta**

**\$\_44\_Cdiva**

**\$\_45\_NRqtaConfezione**

**\$\_46\_NRqtaMinOrdinabile**

**\$\_47\_DSunitaMisura**

**\$\_48\_NRqtaMultiplo**

**\$\_49\_CDcategoria**

**\$\_4\_Mmdescrizione**

**\$\_5\_Dskeywords**

**\$\_6\_Dsdescription**

**\$\_7\_FLschedaTecnica**

**\$\_8\_DSschedaTecnica**

## **`$_9_FLimgDettaglio`**

---

The documentation for this class was generated from the following file:

- `lib/ERP.class.php`

## ERP\_TSD\_ANAGRAFICHE Class Reference

### Data Fields

- `$_0_CDanag`
- `$_1_Cdtipo_anag`
- `$_2_Cdute`
- `$_3_Dsraso`
- `$_4_Dscognome`
- `$_5_Dsnome`
- `$_6_Dsindi`
- `$_7_Dsloca`
- `$_8_Dscap`
- `$_9_Dsprov`
- `$_10_Dsnazione`
- `$_11_Dstel`
- `$_12_Dsfax`
- `$_13_Dscell`
- `$_14_Dsemail`
- `$_15_Dssito`
- `$_16_Dspiva`
- `$_17_Dscod_fisc`
- `$_18_Dsbanca`
- `$_19_Dsabi`
- `$_20_Dscab`
- `$_21_Cdzona`
- `$_22_Dsfonte`
- `$_23_Mmnota`
- `$_24_Fgprivacy`
- `$_25_Cdstato_anag`
- `$_26_Cdfonte`
- `$_27_Fgfiglio`
- `$_28_Fgfittizio`
- `$_29_Dtins`
- `$_30_Cdute_ins`
- `$_31_Dtmod`
- `$_32_Cdute_mod`
- `$_33_Fgeliminato`
- `$_34_Dsluogonascita`
- `$_35_Dssesso`
- `$_36_Fginvia_sms`
- `$_37_Fginvia_email`
- `$_38_Dtnascita`
- `$_39_Dssconto`
- `$_40_CDcatCliente`
- `$_41_Cdiva`
- `$_42_Nrfido`
- `$_43_Nresposizione`
- `$_44_Dsblocco`
- `$_45_Cdlistino`
- `$_46_FGspeseSped`
- `$_47_NRsogliaSpeseSped`
- `$_48_NRcostoFissoSpeseSped`

- **\$\_49\_NRcostoPercSpeseSped**
  - **\$\_50\_CDdest\_pre**
  - **\$\_51\_CDsped\_pre**
  - **\$\_52\_CDpag\_pre**
-

## Field Documentation

**\$\_0\_CDanag**

**\$\_10\_Dsnazione**

**\$\_11\_Dstel**

**\$\_12\_Dsfax**

**\$\_13\_Dscell**

**\$\_14\_Dsemail**

**\$\_15\_Dssito**

**\$\_16\_Dspiva**

**\$\_17\_Dscod\_fisc**

**\$\_18\_Dsbanca**

**\$\_19\_Dsabi**

**\$\_1\_Cdtipo\_anag**

**\$\_20\_Dscab**

**\$\_21\_Cdzona**

**\$\_22\_Dsfonte**

**\$\_23\_Mmnota**

**\$\_24\_Fgprivacy**

**\$\_25\_Cdstato\_anag**

**\$\_26\_Cdfonte**

**\$\_27\_Fgfiglio**

**\$\_28\_Fgfittizio**

**\$\_29\_Dtins**

**\$\_2\_Cdute**

**\$\_30\_Cdute\_ins**



**\$\_31\_Dtmod**

**\$\_32\_Cdute\_mod**

**\$\_33\_Fgeliminato**

**\$\_34\_Dsluogonascita**

**\$\_35\_Dspresso**

**\$\_36\_Fginvia\_sms**

**\$\_37\_Fginvia\_email**

**\$\_38\_Dtnascita**

**\$\_39\_Dssconto**

**\$\_3\_Dsraso**

**\$\_40\_CDcatCliente**

**\$\_41\_Cdiva**

**\$\_42\_Nrfido**

**\$\_43\_Nresposizione**

**\$\_44\_Dsblocco**

**\$\_45\_Cdlistino**

**\$\_46\_FGspeseSped**

**\$\_47\_NRsogliaSpeseSped**

**\$\_48\_NRcostoFissoSpeseSped**

**\$\_49\_NRcostoPercSpeseSped**

**\$\_4\_Dscognome**

**\$\_50\_CDdest\_pre**

**\$\_51\_CDsped\_pre**

**\$\_52\_CDpag\_pre**

**\$\_5\_Dsnome**

**`$_6_Dsindi`**

**`$_7_Dsloca`**

**`$_8_Dscap`**

**`$_9_Dsprov`**

---

**The documentation for this class was generated from the following file:**

- **`lib/ERP.class.php`**

## ERP\_TSD\_SCONTI\_SCAGLIONE Class Reference

### Data Fields

- `$_0_Id`
  - `$_1_CDprodotto`
  - `$_2_NRqtaLimite`
  - `$_3_CDanag`
  - `$_4_NRsconto`
- 

### Field Documentation

`$_0_Id`

`$_1_CDprodotto`

`$_2_NRqtaLimite`

`$_3_CDanag`

`$_4_NRsconto`

---

The documentation for this class was generated from the following file:

- `lib/ERP.class.php`

## ERP\_TSD\_UTENTI Class Reference

### Data Fields

- `$_0_Cdute`
  - `$_1_Dsdesc_ute`
  - `$_2_Dslogin`
  - `$_3_DSpwd`
  - `$_4_Idprof`
  - `$_5_Cdlingua`
  - `$_6_Cddivisa`
  - `$_7_Dsemail_ut`
  - `$_8_DSusr_email_ute`
  - `$_9_DSpwd_email_ute`
  - `$_10_DStel_ute`
  - `$_11_FGstato_gen`
  - `$_12_IDpag_homepage`
  - `$_13_CDstato_gen`
  - `$_14_FGconfermato`
  - `$_15_MMnote`
-

## Field Documentation

**`$_0_Cdute`**

**`$_10_DStel_ute`**

**`$_11_FGstato_gen`**

**`$_12_IDpag_homepage`**

**`$_13_CDstato_gen`**

**`$_14_FGconfermato`**

**`$_15_MMnote`**

**`$_1_Dsdesc_ute`**

**`$_2_Dslogin`**

**`$_3_DSpwd`**

**`$_4_Idprof`**

**`$_5_Cdlingua`**

**`$_6_Cddivisa`**

**`$_7_Dsemail_ut`**

**`$_8_DSusr_email_ute`**

**`$_9_DSpwd_email_ute`**

---

The documentation for this class was generated from the following file:

- `lib/ERP.class.php`

# MySQL Class Reference

## Public Member Functions

- **\_\_construct** (\$connect=true, \$database=null, \$server=null, \$username=null, \$password=null, \$charset=null)
- **\_\_destruct** ()
- **AutoInsertUpdate** (\$tableName, \$valuesArray, \$whereArray)
- **BeginningOfSeek** ()
- **Close** ()
- **DeleteRows** (\$tableName, \$whereArray=null)
- **EndOfSeek** ()
- **Error** ()
- **ErrorNumber** ()
- **GetColumnComments** (\$table)
- **GetColumnCount** (\$table="")
- **GetColumnDataType** (\$column, \$table="")
- **GetColumnID** (\$column, \$table="")
- **GetColumnLength** (\$column, \$table="")
- **GetColumnName** (\$columnID, \$table="")
- **GetColumnNames** (\$table="")
- **GetHTML** (\$showCount=true, \$styleTable=null, \$styleHeader=null, \$styleData=null)
- **GetJSON** ()
- **GetLastInsertID** ()
- **GetLastSQL** ()
- **GetTables** ()
- **GetXML** ()
- **HasRecords** (\$sql="")
- **InsertRow** (\$tableName, \$valuesArray)
- **IsConnected** ()
- **Kill** (\$message="")
- **MoveFirst** ()
- **MoveLast** ()
- **Open** (\$database=null, \$server=null, \$username=null, \$password=null, \$charset=null, \$pcon=false)
- **Query** (\$sql)
- **QueryArray** (\$sql, \$resultType=MYSQLI\_BOTH)
- **QuerySingleRow** (\$sql)
- **QuerySingleRowArray** (\$sql, \$resultType=MYSQLI\_BOTH)
- **QuerySingleValue** (\$sql)
- **QueryTimed** (\$sql)
- **Records** ()
- **RecordsArray** (\$resultType=MYSQLI\_BOTH)
- **Release** ()
- **Row** (\$optional\_row\_number=null)
- **RowArray** (\$optional\_row\_number=null, \$resultType=MYSQLI\_BOTH)
- **RowCount** ()
- **Seek** (\$row\_number)
- **SeekPosition** ()
- **SelectDatabase** (\$database, \$charset="")
- **SelectRows** (\$tableName, \$whereArray=null, \$columns=null, \$sortColumns=null, \$sortAscending=true, \$limit=null)
- **SelectTable** (\$tableName)
- **SQLFix** (\$value)
- **TimerDuration** (\$decimals=4)
- **TimerStart** ()

- **TimerStop** ()
- **TransactionBegin** ()
- **TransactionEnd** ()
- **TransactionRollback** ()
- **TruncateTable** (\$tableName)
- **UpdateRows** (\$tableName, \$valuesArray, \$whereArray=null)

## Static Public Member Functions

- static **BuildSQLDelete** (\$tableName, \$whereArray=null)
- static **BuildSQLInsert** (\$tableName, \$valuesArray)
- static **BuildSQLSelect** (\$tableName, \$whereArray=null, \$columns=null, \$sortColumns=null, \$sortAscending=true, \$limit=null)
- static **BuildSQLUpdate** (\$tableName, \$valuesArray, \$whereArray=null)
- static **BuildSQLWhereClause** (\$whereArray)
- static **GetBooleanValue** (\$value)
- static **IsDate** (\$value)
- static **SQLBooleanValue** (\$value, \$trueValue, \$falseValue, \$datatype=self::SQLVALUE\_TEXT)
- static **SQLValue** (\$value, \$datatype=self::SQLVALUE\_TEXT)

## Data Fields

- const **SQLVALUE\_BIT** = "bit"
- const **SQLVALUE\_BOOLEAN** = "boolean"
- const **SQLVALUE\_DATE** = "date"
- const **SQLVALUE\_DATETIME** = "datetime"
- const **SQLVALUE\_NUMBER** = "number"
- const **SQLVALUE\_T\_F** = "t-f"
- const **SQLVALUE\_TEXT** = "text"
- const **SQLVALUE\_TIME** = "time"
- const **SQLVALUE\_Y\_N** = "y-n"
- **\$mysql\_link** = 0
- **\$ThrowExceptions** = false

---

## Constructor & Destructor Documentation

**\_\_construct** ( *\$connect* = true, *\$database* = null, *\$server* = null, *\$username* = null, *\$password* = null, *\$charset* = null)

Constructor: Opens the connection to the database

### Parameters:

boolean	<i>\$connect</i>	(Optional) Auto-connect when object is created
string	<i>\$database</i>	(Optional) Database name
string	<i>\$server</i>	(Optional) Host address
string	<i>\$username</i>	(Optional) User name
string	<i>\$password</i>	(Optional) Password
string	<i>\$charset</i>	(Optional) Character set

**\_\_destruct** ()

Destructor: Closes the connection to the database

---

## Member Function Documentation

### AutoInsertUpdate ( *\$tableName*, *\$valuesArray*, *\$whereArray*)

Automatically does an INSERT or UPDATE depending if an existing record exists in a table

#### Parameters:

string	<i>\$tableName</i>	The name of the table
array	<i>\$valuesArray</i>	An associative array containing the column names as keys and values as data. The values must be SQL ready (i.e. quotes around strings, formatted dates, ect)
array	<i>\$whereArray</i>	An associative array containing the column names as keys and values as data. The values must be SQL ready (i.e. quotes around strings, formatted dates, ect).

#### Returns:

boolean Returns TRUE on success or FALSE on error

### BeginningOfSeek ()

Returns true if the internal pointer is at the beginning of the records

#### Returns:

boolean TRUE if at the first row or FALSE if not

### static BuildSQLDelete ( *\$tableName*, *\$whereArray* = null)[static]

[STATIC] Builds a SQL DELETE statement

#### Parameters:

string	<i>\$tableName</i>	The name of the table
array	<i>\$whereArray</i>	(Optional) An associative array containing the column names as keys and values as data. The values must be SQL ready (i.e. quotes around strings, formatted dates, ect). If not specified then all values in the table are deleted.

#### Returns:

string Returns the SQL DELETE statement

### static BuildSQLInsert ( *\$tableName*, *\$valuesArray*)[static]

[STATIC] Builds a SQL INSERT statement

#### Parameters:

string	<i>\$tableName</i>	The name of the table
array	<i>\$valuesArray</i>	An associative array containing the column names as keys and values as data. The values must be SQL ready (i.e. quotes around strings, formatted dates, ect)

#### Returns:

string Returns a SQL INSERT statement

### static BuildSQLSelect ( *\$tableName*, *\$whereArray* = null, *\$columns* = null, *\$sortColumns* = null, *\$sortAscending* = true, *\$limit* = null)[static]

Builds a simple SQL SELECT statement



**Parameters:**

string	<i>\$tableName</i>	The name of the table
array	<i>\$whereArray</i>	(Optional) An associative array containing the column names as keys and values as data. The values must be SQL ready (i.e. quotes around strings, formatted dates, ect)
	<i>array/string</i>	<i>\$columns</i> (Optional) The column or list of columns to select
	<i>array/string</i>	<i>\$sortColumns</i> (Optional) Column or list of columns to sort by
boolean	<i>\$sortAscending</i>	(Optional) TRUE for ascending; FALSE for descending This only works if <i>\$sortColumns</i> are specified
	<i>integer/string</i>	<i>\$limit</i> (Optional) The limit of rows to return

**Returns:**

string Returns a SQL SELECT statement

**static BuildSQLUpdate ( *\$tableName*, *\$valuesArray*, *\$whereArray* = null)[static]**

[STATIC] Builds a SQL UPDATE statement

**Parameters:**

string	<i>\$tableName</i>	The name of the table
array	<i>\$valuesArray</i>	An associative array containing the column names as keys and values as data. The values must be SQL ready (i.e. quotes around strings, formatted dates, ect)
array	<i>\$whereArray</i>	(Optional) An associative array containing the column names as keys and values as data. The values must be SQL ready (i.e. quotes around strings, formatted dates, ect). If not specified then all values in the table are updated.

**Returns:**

string Returns a SQL UPDATE statement

**static BuildSQLWhereClause ( *\$whereArray*)[static]**

[STATIC] Builds a SQL WHERE clause from an array. If a key is specified, the key is used at the field name and the value as a comparison. If a key is not used, the value is used as the clause.

**Parameters:**

array	<i>\$whereArray</i>	An associative array containing the column names as keys and values as data. The values must be SQL ready (i.e. quotes around strings, formatted dates, ect)
-------	---------------------	--

**Returns:**

string Returns a string containing the SQL WHERE clause

**Close ()**

Close current **MySQL** connection

**Returns:**

object Returns TRUE on success or FALSE on error

**DeleteRows ( \$tableName, \$whereArray = null)**

Deletes rows in a table based on a WHERE filter (can be just one or many rows based on the filter)

**Parameters:**

string	<i>\$tableName</i>	The name of the table
array	<i>\$whereArray</i>	(Optional) An associative array containing the column names as keys and values as data. The values must be SQL ready (i.e. quotes around strings, formatted dates, ect). If not specified then all values in the table are deleted.

**Returns:**

boolean Returns TRUE on success or FALSE on error

**EndOfSeek ()**

Returns true if the internal pointer is at the end of the records

**Returns:**

boolean TRUE if at the last row or FALSE if not

**Error ()**

Returns the last **MySQL** error as text

**Returns:**

string Error text from last known error

**ErrorNumber ()**

Returns the last **MySQL** error as a number

**Returns:**

integer Error number from last known error

**static GetBooleanValue ( \$value)[static]**

[STATIC] Converts any value of any datatype into boolean (true or false)

**Parameters:**

mixed	<i>\$value</i>	Value to analyze for TRUE or FALSE
-------	----------------	------------------------------------

**Returns:**

boolean Returns TRUE or FALSE

**GetColumnComments ( \$table)**

Returns the comments for fields in a table into an array or NULL if the table has not got any fields

**Parameters:**

string	<i>\$table</i>	Table name
--------	----------------	------------

**Returns:**

array An array that contains the column comments

**GetColumnCount ( \$table = "")**

This function returns the number of columns or returns FALSE on error

**Parameters:**

string	<i>\$table</i>	(Optional) If a table name is not specified, the column count is returned from the last query
--------	----------------	---

**Returns:**

integer The total count of columns

**GetColumnDataType ( \$column, \$table = "" )**

This function returns the data type for a specified column. If the column does not exist or no records exist, it returns FALSE

**Parameters:**

string	<i>\$column</i>	Column name or number (first column is 0)
string	<i>\$table</i>	(Optional) If a table name is not specified, the last returned records are used

**Returns:**

string MySQL data (field) type

**GetColumnID ( \$column, \$table = "" )**

This function returns the position of a column

**Parameters:**

string	<i>\$column</i>	Column name
string	<i>\$table</i>	(Optional) If a table name is not specified, the last returned records are used.

**Returns:**

integer Column ID

**GetColumnLength ( \$column, \$table = "" )**

This function returns the field length or returns FALSE on error

**Parameters:**

string	<i>\$column</i>	Column name
string	<i>\$table</i>	(Optional) If a table name is not specified, the last returned records are used.

**Returns:**

integer Field length

**GetColumnName ( \$columnID, \$table = "" )**

This function returns the name for a specified column number. If the index does not exist or no records exist, it returns FALSE

**Parameters:**

string	<i>\$columnID</i>	Column position (0 is the first column)
string	<i>\$table</i>	(Optional) If a table name is not specified, the last returned records are used.

**Returns:**

integer Field Length

**GetColumnNames ( \$table = "" )**

Returns the field names in a table or query in an array

**Parameters:**

string	<i>\$table</i>	(Optional) If a table name is not specified, the last returned records are used
--------	----------------	---

**Returns:**

array An array that contains the column names

**GetHTML ( \$showCount = true, \$styleTable = null, \$styleHeader = null, \$styleData = null)**

This function returns the last query as an HTML table

**Parameters:**

boolean	<i>\$showCount</i>	(Optional) TRUE if you want to show the row count, FALSE if you do not want to show the count
string	<i>\$styleTable</i>	(Optional) Style information for the table
string	<i>\$styleHeader</i>	(Optional) Style information for the header row
string	<i>\$styleData</i>	(Optional) Style information for the cells

**Returns:**

string HTML containing a table with all records listed

**GetJSON ()**

Returns the last query as a JSON document

**Returns:**

string JSON containing all records listed

**GetLastInsertID ()**

Returns the last autonumber ID field from a previous INSERT query

**Returns:**

integer ID number from previous INSERT query

**GetLastSQL ()**

Returns the last SQL statement executed

**Returns:**

string Current SQL query string

**GetTables ()**

This function returns table names from the database into an array. If the database does not contains any tables, the returned value is FALSE

**Returns:**

array An array that contains the table names

**GetXML ()**

Returns the last query as an XML Document

**Returns:**

string XML containing all records listed

**HasRecords ( \$sql = "")**

Determines if a query contains any rows

**Parameters:**

string	<i>\$sql</i>	[Optional] If specified, the query is first executed Otherwise, the last query is used for comparison
--------	--------------	--

**Returns:**

boolean TRUE if records exist, FALSE if not or query error

**InsertRow ( \$tableName, \$valuesArray)**

Inserts a row into a table in the connected database

**Parameters:**

string	<i>\$tableName</i>	The name of the table
array	<i>\$valuesArray</i>	An associative array containing the column names as keys and values as data. The values must be SQL ready (i.e. quotes around strings, formatted dates, ect)

**Returns:**

integer Returns last insert ID on success or FALSE on failure

**IsConnected ()**

Determines if a valid connection to the database exists

**Returns:**

boolean TRUE idf connectect or FALSE if not connected

**static IsDate ( \$value)[static]**

[STATIC] Determines if a value of any data type is a date PHP can convert

**Parameters:**

<i>date/string</i>	<i>\$value</i>
--------------------	----------------

**Returns:**

boolean Returns TRUE if value is date or FALSE if not date

**Kill ( \$message = "")**

Stop executing (die/exit) and show last **MySQL** error message

**MoveFirst ()**

Seeks to the beginning of the records

**Returns:**

boolean Returns TRUE on success or FALSE on error

**MoveLast ()**

Seeks to the end of the records

**Returns:**

boolean Returns TRUE on success or FALSE on error

**Open ( \$database = null, \$server = null, \$username = null, \$password = null, \$charset = null, \$pcon = false)**

Connect to specified **MySQL** server

**Parameters:**

string	<i>\$database</i>	(Optional) Database name
string	<i>\$server</i>	(Optional) Host address
string	<i>\$username</i>	(Optional) User name
string	<i>\$password</i>	(Optional) Password

string	<i>\$charset</i>	(Optional) Character set
boolean	<i>\$pcon</i>	(Optional) Persistent connection

**Returns:**

boolean Returns TRUE on success or FALSE on error

**Query ( *\$sql* )**

Executes the given SQL query and returns the records

**Parameters:**

string	<i>\$sql</i>	The query string should not end with a semicolon
--------	--------------	--

**Returns:**

object PHP 'mysql result' resource object containing the records on SELECT, SHOW, DESCRIBE or EXPLAIN queries and returns; TRUE or FALSE for all others i.e. UPDATE, DELETE, DROP AND FALSE on all errors (setting the local Error message)

**QueryArray ( *\$sql*, *\$resultType* = MYSQLI\_BOTH )**

Executes the given SQL query and returns a multi-dimensional array

**Parameters:**

string	<i>\$sql</i>	The query string should not end with a semicolon
integer	<i>\$resultType</i>	(Optional) The type of array Values can be: MYSQLI_ASSOC, MYSQLI_NUM, MYSQLI_BOTH

**Returns:**

array A multi-dimensional array containing all the data returned from the query or FALSE on all errors

**QuerySingleRow ( *\$sql* )**

Executes the given SQL query and returns only one (the first) row

**Parameters:**

string	<i>\$sql</i>	The query string should not end with a semicolon
--------	--------------	--

**Returns:**

object PHP resource object containing the first row or FALSE if no row is returned from the query

**QuerySingleRowArray ( *\$sql*, *\$resultType* = MYSQLI\_BOTH )**

Executes the given SQL query and returns the first row as an array

**Parameters:**

string	<i>\$sql</i>	The query string should not end with a semicolon
integer	<i>\$resultType</i>	(Optional) The type of array Values can be: MYSQLI_ASSOC, MYSQLI_NUM, MYSQLI_BOTH

**Returns:**

array An array containing the first row or FALSE if no row is returned from the query

**QuerySingleValue ( *\$sql* )**

Executes a query and returns a single value. If more than one row is returned, only the first value in the first column is returned.

**Parameters:**

string	<i>\$sql</i>	The query string should not end with a semicolon
--------	--------------	--

**Returns:**

mixed The value returned or FALSE if no value

**QueryTimed ( \$sql)**

Executes the given SQL query, measures it, and saves the total duration in microseconds

**Parameters:**

string	<i>\$sql</i>	The query string should not end with a semicolon
--------	--------------	--

**Returns:**

object PHP 'mysql result' resource object containing the records on SELECT, SHOW, DESCRIBE or EXPLAIN queries and returns TRUE or FALSE for all others i.e. UPDATE, DELETE, DROP

**Records ()**

Returns the records from the last query

**Returns:**

object PHP 'mysql result' resource object containing the records for the last query executed

**RecordsArray ( \$resultType = MYSQLI\_BOTH)**

Returns all records from last query and returns contents as array or FALSE on error

**Parameters:**

integer	<i>\$resultType</i>	(Optional) The type of array Values can be: MYSQLI_ASSOC, MYSQLI_NUM, MYSQLI_BOTH
---------	---------------------	---

**Returns:**

Records in array form

**Release ()**

Frees memory used by the query results and returns the function result

**Returns:**

boolean Returns TRUE on success or FALSE on failure

**Row ( \$optional\_row\_number = null)**

Reads the current row and returns contents as a PHP object or returns false on error

**Parameters:**

integer	<i>\$optional_row_number</i>	(Optional) Use to specify a row
---------	------------------------------	---------------------------------

**Returns:**

object PHP object or FALSE on error

**RowArray ( \$optional\_row\_number = null, \$resultType = MYSQLI\_BOTH)**

Reads the current row and returns contents as an array or returns false on error

**Parameters:**

integer	<i>\$optional_row_number</i>	(Optional) Use to specify a row
integer	<i>\$resultType</i>	(Optional) The type of array Values can be: MYSQLI_ASSOC, MYSQLI_NUM, MYSQLI_BOTH

**Returns:**

array Array that corresponds to fetched row or FALSE if no rows

**RowCount ()**

Returns the last query row count

**Returns:**

integer Row count or FALSE on error

**Seek ( \$row\_number)**

Sets the internal database pointer to the specified row number and returns the result

**Parameters:**

integer	<i>\$row_number</i>	Row number
---------	---------------------	------------

**Returns:**

object Fetched row as PHP object

**SeekPosition ()**

Returns the current cursor row location

**Returns:**

integer Current row number

**SelectDatabase ( \$database, \$charset = "")**

Selects a different database and character set

**Parameters:**

string	<i>\$database</i>	Database name
string	<i>\$charset</i>	(Optional) Character set (i.e. utf8)

**Returns:**

boolean Returns TRUE on success or FALSE on error

**SelectRows ( \$tableName, \$whereArray = null, \$columns = null, \$sortColumns = null, \$sortAscending = true, \$limit = null)**

Gets rows in a table based on a WHERE filter

**Parameters:**

string	<i>\$tableName</i>	The name of the table
array	<i>\$whereArray</i>	(Optional) An associative array containing the column names as keys and values as data. The values must be SQL ready (i.e. quotes around strings, formatted dates, ect)
	<i>array/string</i>	<i>\$columns</i> (Optional) The column or list of columns to select
	<i>array/string</i>	<i>\$sortColumns</i> (Optional) Column or list of columns to sort by
boolean	<i>\$sortAscending</i>	(Optional) TRUE for ascending; FALSE for descending This only works if <i>\$sortColumns</i> are specified
	<i>integer/string</i>	<i>\$limit</i> (Optional) The limit of rows to return

**Returns:**

boolean Returns records on success or FALSE on error

**SelectTable ( \$tableName)**

Retrieves all rows in a specified table



**Parameters:**

string	<i>\$tableName</i>	The name of the table
--------	--------------------	-----------------------

**Returns:**

boolean Returns records on success or FALSE on error

**static SQLBooleanValue ( \$value, \$trueValue, \$falseValue, \$datatype = self::SQLVALUE\_TEXT)[static]**

[STATIC] Converts a boolean into a formatted TRUE or FALSE value of choice

**Parameters:**

mixed	<i>\$value</i>	value to analyze for TRUE or FALSE
mixed	<i>\$trueValue</i>	value to use if TRUE
mixed	<i>\$falseValue</i>	value to use if FALSE
string	<i>\$datatype</i>	Use SQLVALUE constants or the strings: string, text, varchar, char, boolean, bool, Y-N, T-F, bit, date, datetime, time, integer, int, number, double, float

**Returns:**

string SQL formatted value of the specified data type

**SQLFix ( \$value)**

Returns string suitable for SQL NOTE: This is no longer a static call because of a conflict with mysqli\_real\_escape\_string. Also, please note that the SQLUnfix function was removed for compatibility.

**Parameters:**

string	<i>\$value</i>	
--------	----------------	--

**Returns:**

string SQL formatted value

**static SQLValue ( \$value, \$datatype = self::SQLVALUE\_TEXT)[static]**

[STATIC] Formats any value into a string suitable for SQL statements (NOTE: Also supports data types returned from the gettype function)

**Parameters:**

mixed	<i>\$value</i>	Any value of any type to be formatted to SQL
string	<i>\$datatype</i>	Use SQLVALUE constants or the strings: string, text, varchar, char, boolean, bool, Y-N, T-F, bit, date, datetime, time, integer, int, number, double, float

**Returns:**

string

**TimerDuration ( \$decimals = 4)**

Returns last measured duration (time between TimerStart and TimerStop)

**Parameters:**

integer	<i>\$decimals</i>	(Optional) The number of decimal places to show
---------	-------------------	---

**Returns:**

Float Microseconds elapsed

**TimerStart ()**

Starts time measurement (in microseconds)

**TimerStop ()**

Stops time measurement (in microseconds)

**TransactionBegin ()**

Starts a transaction

**Returns:**

boolean Returns TRUE on success or FALSE on error

**TransactionEnd ()**

Ends a transaction and commits the queries

**Returns:**

boolean Returns TRUE on success or FALSE on error

**TransactionRollback ()**

Rolls the transaction back

**Returns:**

boolean Returns TRUE on success or FALSE on failure

**TruncateTable ( \$tableName)**

Truncates a table removing all data

**Parameters:**

string	<i>\$tableName</i>	The name of the table
--------	--------------------	-----------------------

**Returns:**

boolean Returns TRUE on success or FALSE on error

**UpdateRows ( \$tableName, \$valuesArray, \$whereArray = null)**

Updates rows in a table based on a WHERE filter (can be just one or many rows based on the filter)

**Parameters:**

string	<i>\$tableName</i>	The name of the table
array	<i>\$valuesArray</i>	An associative array containing the column names as keys and values as data. The values must be SQL ready (i.e. quotes around strings, formatted dates, ect)
array	<i>\$whereArray</i>	(Optional) An associative array containing the column names as keys and values as data. The values must be SQL ready (i.e. quotes around strings, formatted dates, ect). If not specified then all values in the table are updated.

**Returns:**

boolean Returns TRUE on success or FALSE on error

---

## Field Documentation

`$mysql_link = 0`

`$ThrowExceptions = false`

`const SQLVALUE_BIT = "bit"`

`const SQLVALUE_BOOLEAN = "boolean"`

`const SQLVALUE_DATE = "date"`

`const SQLVALUE_DATETIME = "datetime"`

`const SQLVALUE_NUMBER = "number"`

`const SQLVALUE_T_F = "t-f"`

`const SQLVALUE_TEXT = "text"`

`const SQLVALUE_TIME = "time"`

`const SQLVALUE_Y_N = "y-n"`

---

The documentation for this class was generated from the following file:

- `lib/mysql.class.php`

# PHPMailer Class Reference

## Public Member Functions

- **\_\_construct** (\$exceptions=false)
- **IsHTML** (\$ishtml=true)
- **IsSMTP** ()
- **IsMail** ()
- **IsSendmail** ()
- **IsQmail** ()
- **AddAddress** (\$address, \$name= "")
- **AddCC** (\$address, \$name= "")
- **AddBCC** (\$address, \$name= "")
- **AddReplyTo** (\$address, \$name= "")
- **SetFrom** (\$address, \$name= "", \$auto=1)
- **Send** ()
- **PreSend** ()
- **PostSend** ()
- **SmtpConnect** ()
- **SmtpClose** ()
- **SetLanguage** (\$langcode= 'en', \$lang\_path= 'language/')
- **GetTranslations** ()
- **AddrAppend** (\$type, \$addr)
- **AddrFormat** (\$addr)
- **WrapText** (\$message, \$length, \$qp\_mode=false)
- **UTF8CharBoundary** (\$encodedText, \$maxLength)
- **SetWordWrap** ()
- **CreateHeader** ()
- **GetMailMIME** ()
- **GetSentMIMEMessage** ()
- **CreateBody** ()
- **HeaderLine** (\$name, \$value)
- **TextLine** (\$value)
- **AddAttachment** (\$path, \$name= "", \$encoding= 'base64', \$type= 'application/octet-stream')
- **GetAttachments** ()
- **EncodeString** (\$str, \$encoding= 'base64')
- **EncodeHeader** (\$str, \$position= 'text')
- **HasMultiBytes** (\$str)
- **Base64EncodeWrapMB** (\$str, \$lf=null)
- **EncodeQPphp** (\$input= "", \$line\_max=76, \$space\_conv=false)
- **EncodeQP** (\$string, \$line\_max=76, \$space\_conv=false)
- **EncodeQ** (\$str, \$position= 'text')
- **AddStringAttachment** (\$string, \$filename, \$encoding= 'base64', \$type= 'application/octet-stream')
- **AddEmbeddedImage** (\$path, \$cid, \$name= "", \$encoding= 'base64', \$type= 'application/octet-stream')
- **AddStringEmbeddedImage** (\$string, \$cid, \$name= "", \$encoding= 'base64', \$type= 'application/octet-stream')
- **InlineImageExists** ()
- **AttachmentExists** ()
- **AlternativeExists** ()
- **ClearAddresses** ()
- **ClearCCs** ()
- **ClearBCCs** ()
- **ClearReplyTos** ()
- **ClearAllRecipients** ()

- **ClearAttachments** ()
- **ClearCustomHeaders** ()
- **IsError** ()
- **FixEOL** (\$str)
- **AddCustomHeader** (\$name, \$value=null)
- **MsgHTML** (\$message, \$basedir= "")
- **set** (\$name, \$value= "")
- **SecureHeader** (\$str)
- **Sign** (\$cert\_filename, \$key\_filename, \$key\_pass)
- **DKIM\_QP** (\$txt)
- **DKIM\_Sign** (\$s)
- **DKIM\_HeaderC** (\$s)
- **DKIM\_BodyC** (\$body)
- **DKIM\_Add** (\$headers\_line, \$subject, \$body)

## Static Public Member Functions

- static **ValidateAddress** (\$address)
- static **RFCDate** ()
- static **\_mime\_types** (\$ext= "")

## Data Fields

- **\$Priority** = 3
- **\$CharSet** = 'iso-8859-1'
- **\$ContentType** = 'text/plain'
- **\$Encoding** = '8bit'
- **\$ErrorInfo** = ""
- **\$From** = 'root@localhost'
- **\$FromName** = 'Root User'
- **\$Sender** = ""
- **\$ReturnPath** = ""
- **\$Subject** = ""
- **\$Body** = ""
- **\$AltBody** = ""
- **\$WordWrap** = 0
- **\$Mailer** = 'mail'
- **\$Sendmail** = '/usr/sbin/sendmail'
- **\$UseSendmailOptions** = true
- **\$PluginDir** = ""
- **\$ConfirmReadingTo** = ""
- **\$Hostname** = ""
- **\$MessageID** = ""
- **\$MessageDate** = ""
- **\$Host** = 'localhost'
- **\$Port** = 25
- **\$Helo** = ""
- **\$SMTPSecure** = ""
- **\$SMTPAuth** = false
- **\$Username** = ""
- **\$Password** = ""
- **\$AuthType** = ""
- **\$Realm** = ""
- **\$Workstation** = ""
- **\$Timeout** = 10

- **\$SMTPDebug** = false
- **\$Debugoutput** = "echo"
- **\$SMTPKeepAlive** = false
- **\$SingleTo** = false
- **\$SingleToArray** = array()
- **\$LE** = "\n"
- **\$DKIM\_selector** = "
- **\$DKIM\_identity** = "
- **\$DKIM\_passphrase** = "
- **\$DKIM\_domain** = "
- **\$DKIM\_private** = "
- **\$action\_function** = "
- **\$Version** = '5.2.4'
- **\$XMailer** = "
- const **STOP\_MESSAGE** = 0
- const **STOP\_CONTINUE** = 1
- const **STOP\_CRITICAL** = 2
- const **CRLF** = "\r\n"

## Protected Member Functions

- **AddAnAddress** (\$kind, \$address, \$name= "")
- **SendmailSend** (\$header, \$body)
- **MailSend** (\$header, \$body)
- **SmtplibSend** (\$header, \$body)
- **GetBoundary** (\$boundary, \$charSet, \$contentType, \$encoding)
- **EndBoundary** (\$boundary)
- **SetMessageType** ()
- **AttachAll** (\$disposition\_type, \$boundary)
- **EncodeFile** (\$path, \$encoding= 'base64')
- **SetError** (\$msg)
- **ServerHostname** ()
- **Lang** (\$key)
- **doCallback** (\$isSent, \$to, \$cc, \$bcc, \$subject, \$body, \$from=null)

## Protected Attributes

- **\$MIMEBody** = "
- **\$MIMEHeader** = "
- **\$mailHeader** = "
- **\$smtp** = null
- **\$to** = array()
- **\$cc** = array()
- **\$bcc** = array()
- **\$ReplyTo** = array()
- **\$all\_recipients** = array()
- **\$attachment** = array()
- **\$CustomHeader** = array()
- **\$message\_type** = "
- **\$boundary** = array()
- **\$language** = array()
- **\$error\_count** = 0
- **\$sign\_cert\_file** = "
- **\$sign\_key\_file** = "
- **\$sign\_key\_pass** = "

- `$exceptions = false`

---

## Constructor & Destructor Documentation

**`__construct ( $exceptions = false)`**

Constructor

**Parameters:**

boolean	<i>\$exceptions</i>	Should we throw external exceptions?
---------	---------------------	--------------------------------------

---

## Member Function Documentation

**`static _mime_types ( $ext = '')[static]`**

Gets the MIME type of the embedded or inline image

**Parameters:**

string	<i>\$ext</i>	File extension    public
--------	--------------	--------------------------

**Returns:**

string MIME type of ext

**`AddAddress ( $address, $name = '')`**

Adds a "To" address.

**Parameters:**

string	<i>\$address</i>	
string	<i>\$name</i>	

**Returns:**

boolean true on success, false if address already used

**`AddAnAddress ( $kind, $address, $name = '')[protected]`**

Adds an address to one of the recipient arrays Addresses that have been added already return false, but do not throw exceptions

**Parameters:**

string	<i>\$kind</i>	One of 'to', 'cc', 'bcc', 'ReplyTo'
string	<i>\$address</i>	The email address to send to
string	<i>\$name</i>	

**Exceptions:**

<i>phpmailerException</i>	
---------------------------	--

**Returns:**

boolean true on success, false if address already used or invalid in some way    protected

**`AddAttachment ( $path, $name = '', $encoding = 'base64', $type = 'application/octet-stream')`**

Adds an attachment from a path on the filesystem. Returns false if the file could not be found or accessed.

**Parameters:**

string	<i>\$path</i>	Path to the attachment.
--------	---------------	-------------------------

string	<i>\$name</i>	Overrides the attachment name.
string	<i>\$encoding</i>	File encoding (see <i>\$Encoding</i> ).
string	<i>\$type</i>	File extension (MIME) type.

**Exceptions:**

<i>phpmailerException</i>	
---------------------------	--

**Returns:**

bool

**AddBCC ( *\$address*, *\$name* = '' )**

Adds a "Bcc" address. Note: this function works with the SMTP mailer on win32, not with the "mail" mailer.

**Parameters:**

string	<i>\$address</i>	
string	<i>\$name</i>	

**Returns:**

boolean true on success, false if address already used

**AddCC ( *\$address*, *\$name* = '' )**

Adds a "Cc" address. Note: this function works with the SMTP mailer on win32, not with the "mail" mailer.

**Parameters:**

string	<i>\$address</i>	
string	<i>\$name</i>	

**Returns:**

boolean true on success, false if address already used

**AddCustomHeader ( *\$name*, *\$value* = null )**

Adds a custom header. *\$name* value can be overloaded to contain both header name and value (name:value)  
public

**Parameters:**

string	<i>\$name</i>	custom header name
string	<i>\$value</i>	header value

**Returns:**

void

**AddEmbeddedImage ( *\$path*, *\$cid*, *\$name* = '', *\$encoding* = 'base64', *\$type* = 'application/octet-stream' )**

Adds an embedded attachment. This can include images, sounds, and just about any other document. Make sure to set the *\$type* to an image type. For JPEG images use "image/jpeg" and for GIF images use "image/gif".

**Parameters:**

string	<i>\$path</i>	Path to the attachment.
string	<i>\$cid</i>	Content ID of the attachment. Use this to identify the Id for accessing the image in an HTML form.
string	<i>\$name</i>	Overrides the attachment name.
string	<i>\$encoding</i>	File encoding (see <i>\$Encoding</i> ).
string	<i>\$type</i>	File extension (MIME) type.



**Returns:**

bool

**AddrAppend ( \$type, \$addr)**

Creates recipient headers. public

**Parameters:**

string	<i>\$type</i>	
array	<i>\$addr</i>	

**Returns:**

string

**AddReplyTo ( \$address, \$name = '' )**

Adds a "Reply-to" address.

**Parameters:**

string	<i>\$address</i>	
string	<i>\$name</i>	

**Returns:**

boolean

**AddrFormat ( \$addr)**

Formats an address correctly. public

**Parameters:**

string	<i>\$addr</i>	
--------	---------------	--

**Returns:**

string

**AddStringAttachment ( \$string, \$filename, \$encoding = 'base64', \$type = 'application/octet-stream')**

Adds a string or binary attachment (non-filesystem) to the list. This method can be used to attach ascii or binary data, such as a BLOB record from a database.

**Parameters:**

string	<i>\$string</i>	String attachment data.
string	<i>\$filename</i>	Name of the attachment.
string	<i>\$encoding</i>	File encoding (see <i>\$Encoding</i> ).
string	<i>\$type</i>	File extension (MIME) type.

**Returns:**

void

**AddStringEmbeddedImage ( \$string, \$cid, \$name = '', \$encoding = 'base64', \$type = 'application/octet-stream')**

Adds an embedded stringified attachment. This can include images, sounds, and just about any other document. Make sure to set the \$type to an image type. For JPEG images use "image/jpeg" and for GIF images use "image/gif".

**Parameters:**

string	<i>\$string</i>	The attachment.
string	<i>\$cid</i>	Content ID of the attachment. Use this to identify the Id for accessing the image in an HTML form.

string	<i>\$name</i>	Overrides the attachment name.
string	<i>\$encoding</i>	File encoding (see <i>\$Encoding</i> ).
string	<i>\$type</i>	File extension (MIME) type.

**Returns:**

bool

**AlternativeExists ()**

Does this message have an alternative body set?

**Returns:**

bool

**AttachAll ( *\$disposition\_type*, *\$boundary*)[*protected*]**

Attaches all fs, string, and binary attachments to the message. Returns an empty string on failure. *protected*

**Parameters:**

string	<i>\$disposition_type</i>	
string	<i>\$boundary</i>	

**Returns:**

string

**AttachmentExists ()**

Returns true if an attachment (non-inline) is present.

**Returns:**

bool

**Base64EncodeWrapMB ( *\$str*, *\$lf = null*)**

Correctly encodes and wraps long multibyte strings for mail headers without breaking lines within a character. Adapted from a function by paravoid at <http://uk.php.net/manual/en/function.mb-encode-mimeheader.php> public

**Parameters:**

string	<i>\$str</i>	multi-byte text to wrap encode
string	<i>\$lf</i>	string to use as linefeed/end-of-line

**Returns:**

string

**ClearAddresses ()**

Clears all recipients assigned in the TO array. Returns void.

**Returns:**

void

**ClearAllRecipients ()**

Clears all recipients assigned in the TO, CC and BCC array. Returns void.

**Returns:**

void

**ClearAttachments ()**

Clears all previously set filesystem, string, and binary attachments. Returns void.

**Returns:**

void

**ClearBCCs ()**

Clears all recipients assigned in the BCC array. Returns void.

**Returns:**

void

**ClearCCs ()**

Clears all recipients assigned in the CC array. Returns void.

**Returns:**

void

**ClearCustomHeaders ()**

Clears all custom headers. Returns void.

**Returns:**

void

**ClearReplyTos ()**

Clears all recipients assigned in the ReplyTo array. Returns void.

**Returns:**

void

**CreateBody ()**

Assembles the message body. Returns an empty string on failure. public

**Exceptions:**

<i>phpmailerException</i>	
---------------------------	--

**Returns:**

string The assembled message body

**CreateHeader ()**

Assembles message header. public

**Returns:**

string The assembled header

**DKIM\_Add ( \$headers\_line, \$subject, \$body)**

Create the DKIM header, body, as new header

public

**Parameters:**

string	<i>\$headers_line</i>	Header lines
string	<i>\$subject</i>	Subject
string	<i>\$body</i>	Body

**Returns:**

string

### DKIM\_BodyC ( *\$body*)

Generate DKIM Canonicalization Body

public

#### Parameters:

string	<i>\$body</i>	Message Body
--------	---------------	--------------

#### Returns:

string

### DKIM\_HeaderC ( *\$s*)

Generate DKIM Canonicalization Header

public

#### Parameters:

string	<i>\$s</i>	Header
--------	------------	--------

#### Returns:

string

### DKIM\_QP ( *\$txt*)

Set the private key file and password to sign the message.

public

#### Parameters:

string	<i>\$txt</i>	
--------	--------------	--

#### Returns:

string

### DKIM\_Sign ( *\$s*)

Generate DKIM signature

public

#### Parameters:

string	<i>\$s</i>	Header
--------	------------	--------

#### Returns:

string

### doCallback ( *\$isSent*, *\$to*, *\$cc*, *\$bcc*, *\$subject*, *\$body*, *\$from = null*)[protected]

Perform callback

#### Parameters:

boolean	<i>\$isSent</i>	
string	<i>\$to</i>	
string	<i>\$cc</i>	
string	<i>\$bcc</i>	
string	<i>\$subject</i>	
string	<i>\$body</i>	
string	<i>\$from</i>	

### EncodeFile ( *\$path*, *\$encoding = 'base64'*)[protected]

Encodes attachment in requested format. Returns an empty string on failure.

**Parameters:**

string	<i>\$path</i>	The full path to the file
string	<i>\$encoding</i>	The encoding to use; one of 'base64', '7bit', '8bit', 'binary', 'quoted-printable'

**Exceptions:**

<i>phpmailerException</i>	
---------------------------	--

**See also:**

**EncodeFile()** protected

**Returns:**

string

**EncodeHeader ( \$str, \$position = 'text')**

Encode a header string to best (shortest) of Q, B, quoted or none. public

**Parameters:**

string	<i>\$str</i>	
string	<i>\$position</i>	

**Returns:**

string

**EncodeQ ( \$str, \$position = 'text')**

Encode string to q encoding. string \$str the text to encode string \$position Where the text is going to be used, see the RFC for what that means public string

**EncodeQP ( \$string, \$line\_max = 76, \$space\_conv = false)**

Encode string to RFC2045 (6.7) quoted-printable format Uses a PHP5 stream filter to do the encoding about 64x faster than the old version Also results in same content as you started with after decoding

**See also:**

**EncodeQPphp()** public

**Parameters:**

string	<i>\$string</i>	the text to encode
integer	<i>\$line_max</i>	Number of chars allowed on a line before wrapping
boolean	<i>\$space_conv</i>	Dummy param for compatibility with existing EncodeQP function

**Returns:**

string

**Author:**

Marcus Bointon

**EncodeQPphp ( \$input = '', \$line\_max = 76, \$space\_conv = false)**

Encode string to quoted-printable. Only uses standard PHP, slow, but will always work public

**Parameters:**

string	<i>\$input</i>	
integer	<i>\$line_max</i>	Number of chars allowed on a line before wrapping
bool	<i>\$space_conv</i>	

**EncodeString ( \$str, \$encoding = 'base64 ')**

Encodes string to requested format. Returns an empty string on failure.

**Parameters:**

string	<i>\$str</i>	The text to encode
string	<i>\$encoding</i>	The encoding to use; one of 'base64', '7bit', '8bit', 'binary', 'quoted-printable' public

**Returns:**

string

**EndBoundary ( \$boundary)[protected]**

Returns the end of a message boundary. protected

**Parameters:**

string	<i>\$boundary</i>	
--------	-------------------	--

**Returns:**

string

**FixEOL ( \$str)**

Changes every end of line from CRLF, CR or LF to \$this->LE. public

**Parameters:**

string	<i>\$str</i>	String to FixEOL
--------	--------------	------------------

**Returns:**

string

**GetAttachments ()**

Return the current array of attachments

**Returns:**

array

**GetBoundary ( \$boundary, \$charSet, \$contentType, \$encoding)[protected]**

Returns the start of a message boundary. protected

**Parameters:**

string	<i>\$boundary</i>	
string	<i>\$charSet</i>	
string	<i>\$contentType</i>	
string	<i>\$encoding</i>	

**Returns:**

string

**GetMailMIME ()**

Returns the message MIME. public

**Returns:**

string

**GetSentMIMEMessage ()**

Returns the MIME message (headers and body). Only really valid post **PreSend()**. public

**Returns:**

string

**GetTranslations ()**

Return the current array of language strings

**Returns:**

array

**HasMultiBytes ( \$str)**

Checks if a string contains multibyte characters. public

**Parameters:**

string	<i>\$str</i>	multi-byte text to wrap encode
--------	--------------	--------------------------------

**Returns:**

bool

**HeaderLine ( \$name, \$value)**

Returns a formatted header line. public

**Parameters:**

string	<i>\$name</i>	
string	<i>\$value</i>	

**Returns:**

string

**InlinelImageExists ()**

Returns true if an inline attachment is present. public

**Returns:**

bool

**IsError ()**

Returns true if an error occurred. public

**Returns:**

bool

**IsHTML ( \$ishtml = true)**

Sets message type to HTML.

**Parameters:**

bool	<i>\$ishtml</i>	
------	-----------------	--

**Returns:**

void

**IsMail ()**

Sets Mailer to send message using PHP mail() function.

**Returns:**

void

### IsQmail ()

Sets Mailer to send message using the qmail MTA.

#### Returns:

void

### IsSendmail ()

Sets Mailer to send message using the \$Sendmail program.

#### Returns:

void

### IsSMTP ()

Sets Mailer to send message using SMTP.

#### Returns:

void

### Lang ( \$key)[protected]

Returns a message in the appropriate language. protected

#### Parameters:

string	\$key	
--------	-------	--

#### Returns:

string

### MailSend ( \$header, \$body)[protected]

Sends mail using the PHP mail() function.

#### Parameters:

string	\$header	The message headers
string	\$body	The message body

#### Exceptions:

<i>phpmailerException</i>	protected
---------------------------	-----------

#### Returns:

bool

### MsgHTML ( \$message, \$basedir = ' ')

Evaluates the message and returns modifications for inline images and backgrounds public

#### Parameters:

string	\$message	Text to be HTML modified
string	\$basedir	baseline directory for path

#### Returns:

string \$message

### PostSend ()

Actual Email transport function Send the email via the selected mechanism

#### Exceptions:

<i>phpmailerException</i>	
---------------------------	--



<i>on</i>	
-----------	--

**Returns:**

bool

**PreSend ()**

Prep mail by constructing all message entities

**Exceptions:**

<i>phpmailerException</i>	
---------------------------	--

**Returns:**

bool

**static RFCDate ()[static]**

Returns the proper RFC 822 formatted date. public

**Returns:**

string

**SecureHeader ( \$str)**

Strips newlines to prevent header injection. public

**Parameters:**

string	<i>\$str</i>	String
--------	--------------	--------

**Returns:**

string

**Send ()**

Creates message and assigns Mailer. If the message is not sent successfully then it returns false. Use the ErrorInfo variable to view description of the error.

**Exceptions:**

<i>phpmailerException</i>	
---------------------------	--

**Returns:**

bool

**SendmailSend ( \$header, \$body)[protected]**

Sends mail using the \$Sendmail program.

**Parameters:**

string	<i>\$header</i>	The message headers
string	<i>\$body</i>	The message body

**Exceptions:**

<i>phpmailerException</i>	protected
---------------------------	-----------

**Returns:**

bool

**ServerHostname ()[protected]**

Returns the server hostname or 'localhost.localdomain' if unknown. protected

**Returns:**

string

**set ( \$name, \$value = ' ' )**

Set (or reset) Class Objects (variables)

Usage Example: \$page-&gt;set('X-Priority', '3');

public

**Parameters:**

string	<i>\$name</i>	Parameter Name
mixed	<i>\$value</i>	Parameter Value NOTE: will not work with arrays, there are no arrays to set/reset

**Exceptions:**

<i>phpmailerException</i>	
---------------------------	--

**Returns:**

bool

**Todo:**

Should this not be using \_\_set() magic function?

**SetError ( \$msg)[protected]**

Adds the error message to the error container. protected

**Parameters:**

string	<i>\$msg</i>	
--------	--------------	--

**Returns:**

void

**SetFrom ( \$address, \$name = ' ', \$auto = 1)**

Set the From and FromName properties

**Parameters:**

string	<i>\$address</i>	
string	<i>\$name</i>	
int	<i>\$auto</i>	Also set Reply-To and Sender

**Exceptions:**

<i>phpmailerException</i>	
---------------------------	--

**Returns:**

boolean

**SetLanguage ( \$langcode = 'en', \$lang\_path = 'language/')**

Sets the language for all class error messages. Returns false if it cannot load the language file. The default language is English.

**Parameters:**

string	<i>\$langcode</i>	ISO 639-1 2-character language code (e.g. Portuguese: "br")
string	<i>\$lang_path</i>	Path to the language file directory

**Returns:**

bool public

**SetMessageType () [protected]**

Sets the message type. protected

**Returns:**

void

**SetWordWrap ()**

Set the body wrapping. public

**Returns:**

void

**Sign ( \$cert\_filename, \$key\_filename, \$key\_pass)**

Set the private key file and password to sign the message.

public

**Parameters:**

	<i>\$cert_filename</i>	
string	<i>\$key_filename</i>	Parameter File Name
string	<i>\$key_pass</i>	Password for private key

**SmtpClose ()**

Closes the active SMTP session if one exists.

**Returns:**

void

**SmtpConnect ()**

Initiates a connection to an SMTP server. Returns false if the operation failed. SMTP public

**Exceptions:**

<i>phpmailerException</i>	
---------------------------	--

**Returns:**

bool

**SmtpSend ( \$header, \$body) [protected]**

Sends mail via SMTP using PhpSMTP Returns false if there is a bad MAIL FROM, RCPT, or DATA input.

**Parameters:**

string	<i>\$header</i>	The message headers
string	<i>\$body</i>	The message body

**Exceptions:**

<i>phpmailerException</i>	SMTP protected
---------------------------	----------------

**Returns:**

bool

### TextLine ( *\$value*)

Returns a formatted mail line. public

#### Parameters:

string	<i>\$value</i>	
--------	----------------	--

#### Returns:

string

### UTF8CharBoundary ( *\$encodedText*, *\$maxLength*)

Finds last character boundary prior to maxLength in a utf-8 quoted (printable) encoded string. Original written by Colin Brown. public

#### Parameters:

string	<i>\$encodedText</i>	utf-8 QP text
int	<i>\$maxLength</i>	find last character boundary prior to this length

#### Returns:

int

### static ValidateAddress ( *\$address*)[static]

Check that a string looks roughly like an email address should Static so it can be used without instantiation, public so people can overload Conforms to RFC5322: Uses *correct* regex on which FILTER\_VALIDATE\_EMAIL is based; So why not use FILTER\_VALIDATE\_EMAIL? Because it was broken to not allow **atype** valid addresses :( Some Versions of PHP break on the regex though, likely due to PCRE, so use the older validation method for those users. (<http://php.net/manual/en/pcre.installation.php>) **regex Copyright Michael Rushton 2009-10 | <http://squilloople.com/> | Feel free to use and redistribute this code. But please keep this copyright notice.** string \$address The email address to check boolean public

### WrapText ( *\$message*, *\$length*, *\$qp\_mode = false*)

Wraps message for use with mailers that do not automatically perform wrapping and for quoted-printable. Original written by philippe.

#### Parameters:

string	<i>\$message</i>	The message to wrap
integer	<i>\$length</i>	The line length to wrap to
boolean	<i>\$qp_mode</i>	Whether to run in Quoted-Printable mode public

#### Returns:

string

---

## Field Documentation

**\$action\_function = "**

**\$all\_recipients = array() [protected]**

**\$AltBody = "**

**\$attachment = array() [protected]**

**\$AuthType = "**

**\$bcc = array() [protected]**

**\$Body = "**

**\$boundary = array() [protected]**

**\$cc = array() [protected]**

**\$CharSet = 'iso-8859-1'**

**\$ConfirmReadingTo = "**

**\$ContentType = 'text/plain'**

**\$CustomHeader = array() [protected]**

**\$Debugoutput = "echo"**

**\$DKIM\_domain = "**

**\$DKIM\_identity = "**

**\$DKIM\_passphrase = "**

**\$DKIM\_private = "**

**\$DKIM\_selector = "**

**\$Encoding = '8bit'**

**\$error\_count = 0 [protected]**

**\$ErrorInfo = "**

**\$exceptions = false [protected]**

**\$From = 'root@localhost'**

**\$FromName = 'Root User'**

**\$Helo = "**

**\$Host = 'localhost'**

**\$Hostname = "**

**\$language = array() [protected]**

**\$LE = "\n"**

**\$Mailer = 'mail'**

**\$mailHeader = "[protected]**

**\$message\_type = "[protected]**

**\$MessageDate = "**

**\$MessageID = "**

**\$MIMEBody = "[protected]**

**\$MIMEHeader = "[protected]**

**\$Password = "**

**\$PluginDir = "**

**\$Port = 25**

**\$Priority = 3**

**\$Realm = "**

**\$ReplyTo = array() [protected]**

**\$ReturnPath = "**

**\$Sender = "**

**\$Sendmail = '/usr/sbin/sendmail'**

**\$sign\_cert\_file = "[protected]**

**\$sign\_key\_file = "[protected]**

**\$sign\_key\_pass = "[protected]**

```
$SingleTo = false

$SingleToArray = array()

$smtp = null [protected]

$SMTPAuth = false

$SMTPDebug = false

$SMTPKeepAlive = false

$SMTPSecure = ""

$Subject = ""

$Timeout = 10

$to = array() [protected]

$Username = ""

$UseSendmailOptions = true

$Version = '5.2.4'

$WordWrap = 0

$Workstation = ""

$XMailer = ""

const CRLF = "\r\n"

const STOP_CONTINUE = 1

const STOP_CRITICAL = 2

const STOP_MESSAGE = 0
```

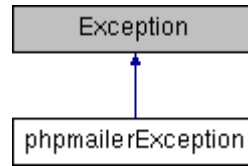
---

The documentation for this class was generated from the following file:

- `lib/class.phpmailer.php`

## phpmailerException Class Reference

Inheritance diagram for phpmailerException:



### Public Member Functions

- `errorMessage ()`

---

### Member Function Documentation

#### `errorMessage ()`

Prettify error message output

#### Returns:

string

---

The documentation for this class was generated from the following file:

- `lib/class.phpmailer.php`



## PS\_specific\_price Class Reference

### Data Fields

- **\$id\_product**
- **\$id\_customer**
- **\$price**
- **\$from\_quantity**
- **\$reduction**
- **\$reduction\_type**
- **\$from**
- **\$to**
- **\$type**
- **\$sconto**
- **\$prezzoNonScontato**

---

### Field Documentation

**\$from**

**\$from\_quantity**

**\$id\_customer**

**\$id\_product**

**\$prezzoNonScontato**

**\$price**

**\$reduction**

**\$reduction\_type**

**\$sconto**

**\$to**

**\$type**

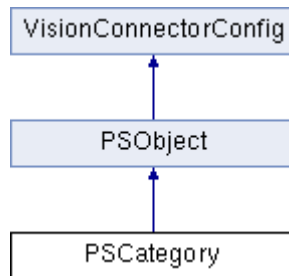
---

The documentation for this class was generated from the following file:

- **lib/ERP.class.php**

## PSCategory Class Reference

Inheritance diagram for PSCategory:



### Public Member Functions

- **\_\_construct ()**
- **getCategory ()**
- **resetCategory ()**
- **getCategoryLang ()**
- **getCategoryIdFromCode (\$code)**
- **updateCategoryLang (\$arrCat, \$id\_category, \$lang)**
- **reenableCategory (\$id\_category)**
- **insertCategory (\$arrCat)**
- **getLeftParent (\$parent)**
- **getIdCategoryParentFromCode (\$code)**
- **preUpdateStructure (\$rgt)**
- **updateCategoryNright (\$nleft)**
- **insertCategoryGroup (\$id\_category)**
- **getGroupId ()**
- **insertCategoryShop (\$id\_category)**
- **getCategoryShopLastPosition ()**
- **insertCategoryLang (\$arrCat, \$id\_category, \$lang)**
- **updateTopMenuCat (\$categories)**
- **insertCategoryImage (\$img, \$id\_category, \$code)**
- **unlinkCategoryImage (\$id\_category)**

### Additional Inherited Members

---

## Constructor & Destructor Documentation

**\_\_construct ()**

---

## Member Function Documentation

**getCategory ()**

Metodo richiamato per la gestione delle Categorie dal connector PHP

**getCategoryIdFromCode ( \$code)**

Restituisce id della categoria a partire dal CODE Vision

### getCategoryLang ()

Metodo utilizzato per recuperare array di lingue a partire dal file passato da ERP

### getCategoryShopLastPosition ()

Metodo utilizzato per recuperare ultimo valore di position all'interno della tabella ps\_category\_shop

#### Returns:

int \$position Ultimo valore di position presente

### getGroupId ()

Metodo che seleziona i gruppi dalla tabelal ps\_group

#### Returns:

Object Lista di strutture contenente id\_group per ogni gruppo

### getIdCategoryParentFromCode ( \$code)

Metodo che restituisce il level\_depth e id category a partire dal codice della categoria

#### Parameters:

type	<i>\$code</i>	Codice ERp della categoria
------	---------------	----------------------------

#### Returns:

Object Struttura custom contente level\_depth e idcategory della categoria selezionata

### getLeftParent ( \$parent)

Metodo utilizzato per restituire nleft della categoria padre

#### Parameters:

int	<i>\$parent</i>	id della categoria genitore
-----	-----------------	-----------------------------

#### Returns:

int nleft della categoria genitore

### insertCategory ( \$arrCat)

Metodo utilizzato per inserire oggetto Categoria passato da ERP all'interno di Prestashop

#### Parameters:

<i>\$arrCat</i>	Array delel categorie recuperato dal file del tracciato passato da ERP	
-----------------	--	--

### insertCategoryGroup ( \$id\_category)

Metodo per inserire la categoria nel gruppo corrispondente (ps\_category\_group)

#### Parameters:

int	<i>\$id_category</i>	Id della categoria
-----	----------------------	--------------------

### insertCategoryImage ( \$img, \$id\_category, \$code)

Metodo utilizzato per inserire l'immagine della categoria se pasata dal tracciato ERP

#### Parameters:

String	<i>\$img</i>	nome immagine da associare alla categoria
Int	<i>\$id_category</i>	Id della categoria
String	<i>\$code</i>	Codice delal categoria passato da ERP

**insertCategoryLang ( \$arrCat, \$id\_category, \$lang)**

Metodo utilizzato per valorizzare le etichette delle categorie in base alle lingue dell'e-commerce e i tracciati passati da ERP

**Parameters:**

type	<i>\$arrCat</i>	Array contenente i dati della categoria ricevuti dal tracciato ERP
type	<i>\$id_category</i>	Id della categoria inserita in Prestashop
type	<i>\$lang</i>	Array delle lingue ricevuto dal tracciato ERP (Se empty vengono utilizzati i valori del tracciato di default)

**insertCategoryShop ( \$id\_category)**

Metodo utilizzato per inserire la categoria all'interno della tabella ps\_category\_shop

**Parameters:**

int	<i>\$id_category</i>	Id della categoria
-----	----------------------	--------------------

**preUpdateStructure ( \$rgt)**

Metodo utilizzato per riallineare la struttura delle categorie secondo la logica nleft / nright "nested set model"

**Parameters:**

int	<i>\$rgt</i>	nright per riallineare la struttura delle categorie
-----	--------------	---

**reenableCategory ( \$id\_Category)**

DEPRECATED Riabilita le categorie precedentemente disattivate

**Parameters:**

<i>\$id_Category</i>	id Prestashop della categoria da abilitare
----------------------	--

**resetCategory ()**

Metodo richiamato in caso di azzeramento e ricarica per fare il reset delle categorie

**unlinkCategoryImage ( \$id\_category)**

Metodo utilizzato per cancellare le immagini già presenti per la categoria

**Parameters:**

Int	<i>\$id_category</i>	Id della categoria
-----	----------------------	--------------------

**updateCategoryLang ( \$arraCat, \$id\_category, \$lang)**

Metodo per aggiornare le etichette di una categoria già esistente Parametri

**Parameters:**

<i>\$arraCat</i>	Array della categoria passato da ERP
<i>\$id_category</i>	Id della categoria già presente nel DB di Prestashop
<i>\$lang</i>	Array contenente le etichette per la categoria

**updateCategoryNright ( \$nleft)**

Metodo utilizzato per riallineare la struttura delle categorie secondo la logica "nested set model"

**Parameters:**

int	<i>\$nleft</i>	Intero nleft
-----	----------------	--------------

### **updateTopMenuCat ( *\$categories* )**

Metodo utilizzato per popolare in automatico il modulo TOPMENU\_HORIZONTAL di Prestashop

#### **Parameters:**

Array	<i>\$categories</i>	Array contente gli id delle categorie da mostrare come voci principali nel top menu horizontal modulo di Prestashop
-------	---------------------	---

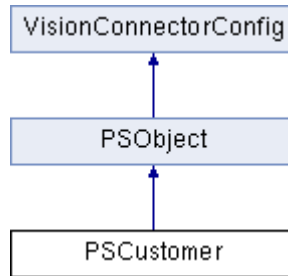
---

**The documentation for this class was generated from the following file:**

- `classes/PSCategory.php`

## PSCustomer Class Reference

Inheritance diagram for PSCustomer:



### Public Member Functions

- **getCustomer ()**
- **getAnagrafiche ()**
- **getEnabledCustomer ()**
- **resetCustomers ()**
- **adjustCutomer ()**
- **existsCustomer (\$code)**
- **insertCustomer (\$arrCst)**
- **updateCustomer (\$arrCst, \$id\_customer)**
- **getCustomerFromId (\$id\_customer)**
- **resetAddressNonFatturazione ()**
- **loadRelAnagrafiche ()**
- **insertAnagrafica (\$anagrafica, \$arrDest)**
- **getIdCountry (\$country)**
- **getIdState (\$id\_country, \$state)**
- **checkIfAnagraficaFatturazioneEsiste (\$vs\_code)**
- **getIdUserFromCode (\$code)**
- **refreshCustomer (\$id\_cust, \$firstname, \$lastname, \$company, \$fido, \$utilizzo)**

### Additional Inherited Members

---

### Detailed Description

Description of **PSCustomer**

**Author:**

ALLINITSRLS

---

### Member Function Documentation

#### **adjustCutomer ()**

Metodo utilizzato per assegnare gruppo di default Customer agli utenti caricati in Prestashop

#### **checkIfAnagraficaFatturazioneEsiste ( \$vs\_code)**

#### **existsCustomer ( \$code)**

Metodo utilizzato per verificare se un utente è già presente in Prestashop a partire dal codice ERP

**Parameters:**

String	<i>\$code</i>	Codice univoco ERP
--------	---------------	--------------------

**Returns:**

int Id dell'utente, 0 se non presente

**getAnagrafiche ()**

Metodo utilizzato per inserire le anagrafiche / indirizzi degli utenti in Prestashop

**Returns:**

type

**getCustomer ()**

Metodo utilizzato per caricare dentro Prestashop tutte le utenze passate da tracciato ERP

**getCustomerFromId ( *\$id\_customer*)****getEnabledCustomer ()**

Metodo utilizzato per recuperare tutti gli utenti attivi in Prestashop

**Returns:**

type

**getIdCountry ( *\$country*)****getIdState ( *\$id\_country*, *\$state*)****getIdUserFromCode ( *\$code*)****insertAnagrafica ( *\$anagrafica*, *\$arrDest*)****insertCustomer ( *\$arrCst*)**

Metodo utilizzato per inserire utente all'interno di prestashop in base ai dati passati dal tracciato ERP

**Parameters:**

Array	<i>\$arrCst</i>	array di dati dell'utente dal tracciato ERP
-------	-----------------	---

**Returns:**

type

**loadRelAnagrafiche ()**

Metodo utilizzato per caricare tutte le anagrafiche

**Returns:**

type Struttura anagrafiche da tracciato ERP

**refreshCustomer ( *\$id\_cust*, *\$firstname*, *\$lastname*, *\$company*, *\$fido*, *\$utilizzo*)****resetAddressNonFatturazione ()****resetCustomers ()**

Metodo per disabilitare gli utenti presenti in Prestashop

### **updateCustomer ( *\$arrCst*, *\$id\_customer* )**

MEtodo utilizzato per fare update di utente esistente

#### **Parameters:**

Array	<i>\$arrCst</i>	Array utente da tracciato ERP
Int	<i>\$id_customer</i>	Id utente esistente

#### **Returns:**

type

---

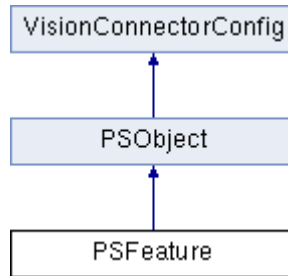
**The documentation for this class was generated from the following file:**

- `classes/PSCustomer.php`



## PSFeature Class Reference

Inheritance diagram for PSFeature:



### Public Member Functions

- **getCaratteristiche ()**
- **reloadForCaratteristiche ()**
- **insertFeatureValue** (\$idCarat, \$idVal, \$featuresGrouped)
- **insertFeatureValueLang** (\$idVal, \$caratVal)
- **insertFeatureProduct** (\$productID, \$data, \$id\_value)
- **insertFeature** (\$data, \$position, \$caratteristicheInserite)
- **getLayeredFilter ()**
- **getCategoryStringForLayeredTemplate ()**
- **insertLayeredCategory** (\$category, \$idValueCarat)

### Additional Inherited Members

---

### Detailed Description

Description of **PSFeature**

**Author:**

ALLINITSRLS

---

## Member Function Documentation

**getCaratteristiche ()**

**getCategoryStringForLayeredTemplate ()**

**getLayeredFilter ()**

**insertFeature ( *\$data*, *\$position*, *\$caratteristicheInserite*)**

**insertFeatureProduct ( *\$productID*, *\$data*, *\$id\_value*)**

**insertFeatureValue ( *\$idCarat*, *\$idVal*, *\$featuresGrouped*)**

**insertFeatureValueLang ( *\$idVal*, *\$caratVal*)**

**insertLayeredCategory ( *\$category*, *\$idValueCarat*)**

**reloadForCaratteristiche ()**

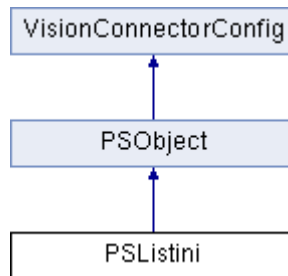
---

The documentation for this class was generated from the following file:

- `classes/PSFeature.php`

## PSListini Class Reference

Inheritance diagram for PSListini:



### Public Member Functions

- **insertGestioneListini** ()
- **mappingUtentiCodeId** ()
- **mappingProdottiCodeId** ()
- **deleteOldGroup** ()
- **createListiniBaseGroup** (\$id\_listino)
- **getCustomerIdFromCode** (\$code)
- **calculateREduction** (\$array\_sconti)
- **insertListiniBaseGroups** (\$specific\_prezzi\_speciali, \$id\_product, \$id\_group)
- **getIdProductFromCode** (\$code)
- **insertErpScontiScaglione** (\$sconto\_scaglione)
- **insertErpPrezziSpeciali** (\$prezzi\_special)
- **checkScontoScaglioneQt** (\$CDProdotto, \$NRqtaLimite, \$prezzi\_special)
- **getAllProductForTaglie** ()
- **getProductPsSpecificPrice** (\$id\_product)
- **insertCombinationSpecificPrice** (\$specific\_price)
- **startFillSpecificPrice** ()
- **checkIfExistsHighPriceRuleCCORRI** (\$specific\_prezzi\_speciali, \$id\_product)
- **checkIfExistsHighPriceRule** (\$specific\_prezzi\_speciali, \$id\_product, \$id\_customer)
- **insertSpecificPriceFromRuleGroup** (\$specific\_prezzi\_speciali, \$id\_product, \$id\_group, \$from, \$to)
- **insertSpecificPriceFromRule** (\$specific\_prezzi\_speciali, \$id\_product, \$id\_customer, \$from, \$to)
- **getPriceListBaseGroupFromPrId** (\$id\_product, \$id\_grp\_listbase)
- **getReductionListBaseGroupFromPrId** (\$id\_product, \$id\_grp\_listbase)
- **getProductBasePrice** (\$id\_product)
- **isTherePriceHighPriority** (\$id\_product, \$id\_customer)
- **insertCustomerGroup** (\$id\_customer, \$id\_group)
- **setDefaultGroup** (\$id\_customer, \$id\_group)
- **enableScontiQt** ()

### Additional Inherited Members

---

### Detailed Description

Description of **PSListini**

**Author:**

ALLINITSRSL

---

## Member Function Documentation

**calculateREduction ( *\$array\_sconti*)**

**checkIfExistsHighPriceRule ( *\$specific\_prezzi\_speciali*, *\$id\_product*, *\$id\_customer*)**

**checkIfExistsHighPriceRuleCCORRI ( *\$specific\_prezzi\_speciali*, *\$id\_product*)**

**checkScontoScaglioneQt ( *\$CDProdotto*, *\$NRqtaLimite*, *\$prezzi\_special*)**

**createListiniBaseGroup ( *\$id\_listino*)**

**deleteOldGroup ()**

**enableScontiQt ()**

**getAllProductForTaglie ()**

**getCustomerIdFromCode ( *\$code*)**

**getIdProductFromCode ( *\$code*)**

**getPriceListBaseGroupFromPrId ( *\$id\_product*, *\$id\_grp\_listbase*)**

**getProductBasePrice ( *\$id\_product*)**

**getProductPsSpecificPrice ( *\$id\_product*)**

**getReductionListBaseGroupFromPrId ( *\$id\_product*, *\$id\_grp\_listbase*)**

**insertCombinationSpecificPrice ( *\$specific\_price*)**

**insertCustomerGroup ( *\$id\_customer*, *\$id\_group*)**

**insertErpPrezziSpeciali ( *\$prezzi\_special*)**

**insertErpScontiScaglione ( *\$sconto\_scaglione*)**

**insertGestioneListini ()**

**insertListiniBaseGroups ( *\$specific\_prezzi\_speciali*, *\$id\_product*, *\$id\_group*)**

**insertSpecificPriceFromRule ( *\$specific\_prezzi\_speciali*, *\$id\_product*, *\$id\_customer*, *\$from*, *\$to*)**

**insertSpecificPriceFromRuleGroup ( *\$specific\_prezzi\_speciali*, *\$id\_product*, *\$id\_group*, *\$from*, *\$to*)**

**isTherePriceHighPriority ( *\$id\_product*, *\$id\_customer*)**

**mappingProdottiCodeId ()**

**mappingUtentiCodeId ()**

**setDefaultGroup ( *\$id\_customer*, *\$id\_group*)**

**startFillSpecificPrice ()**

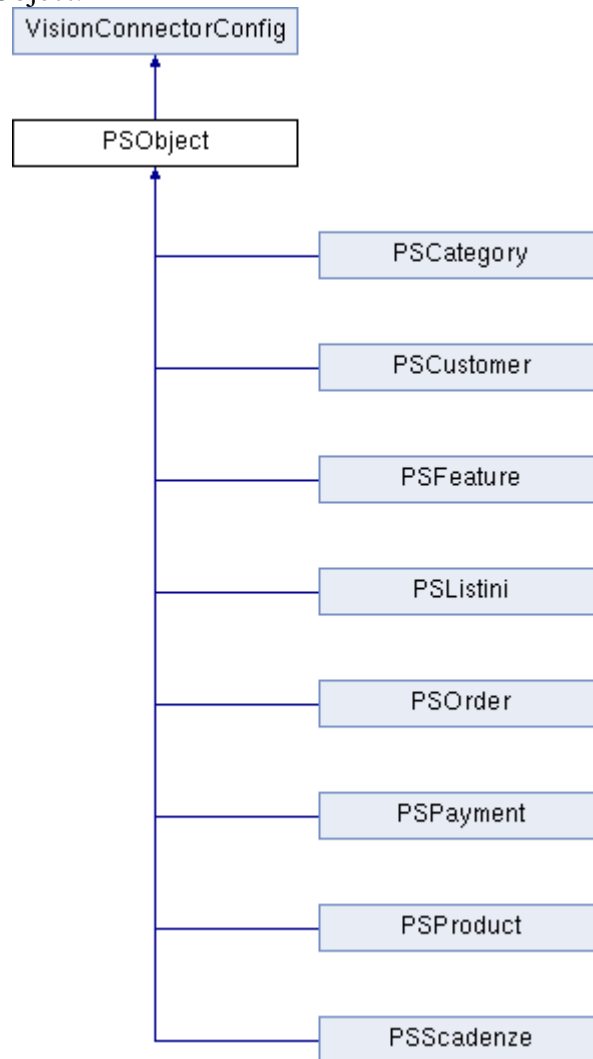
---

**The documentation for this class was generated from the following file:**

- **classes/PSListini.php**

## PSObject Class Reference

Inheritance diagram for PSObject:



### Public Member Functions

- **initImportFolder ()**
- **extractFileIntoFolder ()**
- **checkVersionImporter ()**
- **file\_get\_contents\_utf8 (\$fn)**
- **fileLog (\$message)**
- **sendMailLog ()**
- **checkInsertOrAppend (\$file\_check)**
- **readInfoEcommerce ()**
- **checkInfoExists (\$field\_name)**
- **truncateTable (\$tablename)**
- **removeBOM (\$str="")**
- **slugify (\$text)**
- **regenerateThumbnail ()**
- **checkIfThereAreImages ()**
- **getIdProductFromCode (\$code)**

- `clearPSCache ()`

## Additional Inherited Members

---

### Detailed Description

Description of **PSObject**

**Author:**

ALLINITSRLS

---

### Member Function Documentation

#### **checkIfThereAreImages ()**

Metodo per controllare se sono presenti immagini

**Returns:**

boolean Se TRUE immagini presenti, FALSE non ci sono immagini

#### **checkInfoExists ( *\$field\_name* )**

Metodo utilizzato per controllare se il campo è da aggiornare o inserire nella tabella vsecommerce\_info

**Parameters:**

String	<i>\$field_name</i>	Nome del parametro
--------	---------------------	--------------------

**Returns:**

boolean Se TRUE il campo field\_name esiste nella cartella vsecommerce\_info, falso il campo non esiste

#### **checkInsertOrAppend ( *\$file\_check* )**

Metodo utilizzato per capire se fare AZZERARE E RICARICA sul file (oggetto) in questione

**Parameters:**

String	<i>\$file_check</i>	Nome del file da analizzare
--------	---------------------	-----------------------------

**Returns:**

int Se 0 AZZERARE E RICARICA, 1 in APPEND MODE

#### **checkVersionImporter ()**

Metodo utilizzato per fare il check della versione dei tracciati inviati da ERP

**Returns:**

boolean Se TRUE versione valida, FALSE versione non corretta

#### **clearPSCache ()**

#### **extractFileIntoFolder ()**

Metodo utilizzato per estrarre i tracciati dati e immagini nella cartella import

#### **file\_get\_contents\_utf8 ( *\$fn* )**

Metodo utilizzato per rendere il contenuto di un tracciato ricevuto da ERP utilizzabile dal connector PHP

**Parameters:**

String	<i>\$fn</i>	Percorso del file da leggere
--------	-------------	------------------------------

**Returns:**

String Contenuto del file appena letto in UTF-8 senza BOM

**fileLog ( *\$message*)**

Metodo utilizzato per la creazione del file di log

**Parameters:**

String	<i>\$message</i>	Stringa da inserire nel file di log
--------	------------------	-------------------------------------

**getIdProductFromCode ( *\$code*)****initImportFolder ()**

Metodo utilizzato prima di ogni import per eliminare i file del tracciato e le immagini temporanee generate

**readInfoEcommerce ()**

Metodo utilizzato per leggere le informazioni da ERP e inserirle nella tabella vseccommerce\_info

**regenerateThumbnail ()**

Metodo utilizzato per rigenerare tutte le immagini degli oggetti di Prestahop Utilizzato override interno di classe protected di Prestashop in AdminImagesController

**removeBOM ( *\$str* = "")**

Metodo per rimuovere il carattere di BOM da fine stringa

**Parameters:**

String	<i>\$str</i>	Stringa a cui rimuovere
--------	--------------	-------------------------

**Returns:**

String Stringa rimossa del BOM

**sendMailLog ()**

Metodo utilizzato per inviare email

**slugify ( *\$text*)**

Metodo per codificare una stringa in url corretta

**Parameters:**

String	<i>\$text</i>	Stringa da codificare in url
--------	---------------	------------------------------

**Returns:**

string Stringa in modalità slugify

**truncateTable ( *\$tablename*)**

Metodo per fare il TRUNCATE della tabella passato come parametro

**Parameters:**

String	<i>\$tablename</i>	Nome della tabella
--------	--------------------	--------------------

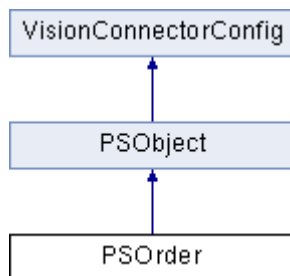


The documentation for this class was generated from the following file:

- `classes/PSObject.php`

## PSOrder Class Reference

Inheritance diagram for PSOrder:



### Public Member Functions

- `getOrderDocuments ()`
- `getOrderDocumentsDetails ()`
- `insertOrderDocuments ($data)`
- `insertOrderDocumentsDetails ($data)`
- `insertOrderDocumentsPdf ()`

### Additional Inherited Members

---

### Detailed Description

Description of **PSOrder**

**Author:**

ALLINITSRLS

---

### Member Function Documentation

`getOrderDocuments ()`

`getOrderDocumentsDetails ()`

`insertOrderDocuments ( $data)`

`insertOrderDocumentsDetails ( $data)`

`insertOrderDocumentsPdf ()`

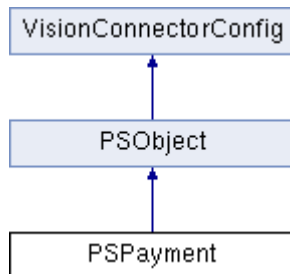
---

The documentation for this class was generated from the following file:

- `classes/PSOrder.php`

## PSPayment Class Reference

Inheritance diagram for PSPayment:



### Public Member Functions

- `getPaymentMethod ()`
- `insertPaymentMethod ($data)`

### Additional Inherited Members

---

### Detailed Description

Description of **PSPayment**

**Author:**

ALLINITSRLS

---

### Member Function Documentation

**getPaymentMethod ()**

**insertPaymentMethod ( *\$data* )**

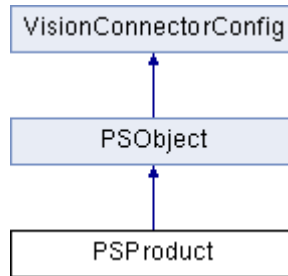
---

The documentation for this class was generated from the following file:

- `classes/PSPayment.php`

## PSPProduct Class Reference

Inheritance diagram for PSPProduct:



### Public Member Functions

- **getProduct ()**
- **getRelatedProducts ()**
- **insertAttributeGroup ()**
- **checkAttributeGroupSet ()**
- **insertAttributeGroupLang (\$id\_attribute\_group)**
- **insertAttributeGroupToShop (\$id\_attribute\_group)**
- **getProductLang ()**
- **existsProduct (\$code)**
- **insertProduct (\$arrPrd, \$code\_product, \$has\_combination)**
- **getCategoryIdFromCode (\$code)**
- **insertCategoryInAllTree (\$idProduct, \$id\_category\_default)**
- **getArrayCatParent (\$id\_category)**
- **insertCategoryProduct (\$idProduct, \$idCategory)**
- **getNleftNrightCategory (\$id\_category)**
- **insertProductLang (\$arrPrd, \$idProduct, \$lang)**
- **insertProductShop (\$arrPrd, \$idProduct)**
- **insertStockProduct (\$arrPrd, \$idProduct)**
- **insertAttachment (\$new, \$id\_product, \$FlschedaTecnica, \$DSschedaTecnica)**
- **newAttachment (\$id\_product, \$file\_name, \$description)**
- **deletePrecAttachment (\$id\_product)**
- **updateNameAttachment (\$id\_product, \$file\_name)**
- **updateDescriptionAttachment (\$id\_product, \$description)**
- **checkAttachment (\$id\_product, \$file\_name, \$description)**
- **updateProduct (\$arrPrd, \$id\_product, \$code\_product, \$has\_combination)**
- **updateProductLang (\$arrPrd, \$idProduct, \$lang)**
- **updateProductShop (\$arrPrd, \$idProduct)**
- **updateStockProduct (\$arrPrd, \$idProduct)**
- **checkIfAttributeExists (\$value)**
- **insertAttribute (\$id\_attribute\_group)**
- **insertAttributeShop (\$id\_attribute)**
- **insertLangAttribute (\$value, \$id\_attribute)**
- **insertAttributeToProduct (\$id\_product\_parent, \$data)**
- **insertValueAttributeShio (\$id\_product\_attribute, \$data, \$id\_product)**
- **insertStockAttrProduct (\$id\_product\_attribute, \$data, \$id\_product)**
- **checkIfAttributeStockExists (\$id\_product\_attribute)**
- **insertProductCombination (\$id\_attribute, \$id\_product\_attribute)**
- **setProductPriceZero (\$id\_product)**
- **adjustSpecifiRule (\$id\_product)**
- **resetImages (\$id\_product)**
- **reinitImageLang (\$id\_image)**

- **reinitImageShop** (\$id\_image)
- **insertProductImage2** (\$img\_1, \$position, \$id\_product)
- **addImageToFolder** (\$idImage)
- **unlinkImageProduct** (\$img\_prd\_pathdefault)
- **checkIfParentExists** (\$imgName)
- **checkIfAlreayExistsGroupPrID** (\$id\_product)
- **insertProduttori** ()
- **getIDManufacturer** (\$produttore)
- **updateManufacturInsideProduct** (\$prod\_code, \$id\_manufacturer)
- **reinitProductImage** (\$id\_product)
- **insertRelatedProduct** (\$data)

## Additional Inherited Members

---

### Detailed Description

Description of **PSProduct**

**Author:**

ALLINITSRLS

---

### Member Function Documentation

#### **addImageToFolder ( \$idImage)**

Metodo per generare la stringa dove copiare le immagini

**Parameters:**

int	<i>\$idImage</i>	Id immagine
-----	------------------	-------------

**Returns:**

String Stringa generata della cartella secondo la logica di Prestashop

#### **adjustSpecifiRule ( \$id\_product)**

Metodo per configurare regola da utilizzare in ps\_specific\_price\_priority

**Parameters:**

Int	<i>\$id_product</i>	Id del prodotto
-----	---------------------	-----------------

#### **checkAttachment ( \$id\_product, \$file\_name, \$description)**

Faccio il check per capire se c'era o meno il file allegato da aggiornare

**Parameters:**

int	<i>\$id_product</i>	Id del prodotto
String	<i>\$file_name</i>	Nome del file allegato
String	<i>\$description</i>	Descrizione del file allegato

#### **checkAttributeGroupSet ()**

Controllo se già configurato attributo gruppo per la gestione delle taglie

**Returns:**

int Id del gruppo

**checkIfAlreayExistsGroupPrID ( \$id\_product)**

Metodo se già esiste il raggruppamento per immagine

**Parameters:**

int	<i>\$id_product</i>	Id del prodotto
-----	---------------------	-----------------

**Returns:**

boolean TRUE se esiste, FALSE non c'è

**checkIfAttributeExists ( \$value)**

Controllo se già esiste attributo, non creo doppione

**Parameters:**

String	<i>\$value</i>	Nome dell'attributo
--------	----------------	---------------------

**Returns:**

int Id attributo se già presente, 0 se non presente

**checkIfAttributeStockExists ( \$id\_product\_attribute)**

Controllo se in stock è già presente attributo

**Parameters:**

Int	<i>\$id_product_attribute</i>	Id attributo del prodotto
-----	-------------------------------	---------------------------

**Returns:**

boolean TRUE se attributo esiste, FALSE non presente

**checkIfParentExits ( \$imgName)**

Metodo per controllare se esiste immagine genitore del prodotto secondo la logica del raggruppamento per immagini

**Parameters:**

String	<i>\$imgName</i>	Nome immagine
--------	------------------	---------------

**Returns:**

int 1 esiste immagine padre, 0 è un'immagine padre

**deletePrecAttachment ( \$id\_product)**

Cancellazione file allegato al prodotto

**Parameters:**

INT	<i>\$id_product</i>	Id del prodotto
-----	---------------------	-----------------

**existsProduct ( \$code)**

Metodo utilizzato per recuperare se esiste o meno il prodotto

**Parameters:**

String	<i>\$code</i>	Codice identificativo del prodotto
--------	---------------	------------------------------------

**Returns:**

int Id del prodotto. Se 0 il prodotto non esiste

**getArrayCatParent ( \$id\_category)**

Metodo utilizzato per l'array delle categorie di appartenenza

**Parameters:**

Int	<i>\$id_category</i>	Id della categoria
-----	----------------------	--------------------

**Returns:**

Array Array di Strutture in ps\_category

**getCategoryIdFromCode ( \$code)**

Restituisce id della categoria a partire dal CODE Vision

**Parameters:**

String	<i>\$code</i>	Codice della categoria ERP
--------	---------------	----------------------------

**Returns:**

int Id della categoria

**getIdManufacturer ( \$produttore)**

Metodo utilizzato per recuperare id del produttore a partire dal nome

**Parameters:**

String	<i>\$produttore</i>	Nome del produttore
--------	---------------------	---------------------

**Returns:**

Int Id del produttore presente nel db

**getNleftNrightCategory ( \$id\_category)**

Metodo utilizzato per recuperare nleft e nright di ogni categoria

**Parameters:**

Int	<i>\$id_category</i>	Id della categoria
-----	----------------------	--------------------

**Returns:**

int

**getProduct ()**

Metodo per la gestione dei prodotti passati da ERP

**Returns:**

type

**getProductLang ()**

Metodo che recupera le etichette dei prodotti nelle varie lingue

**Returns:**

Array Struttura contenente le traduzioni delle etichette dei prodotti

**getRelatedProducts ()****insertAttachment ( \$new, \$id\_product, \$FlschedaTecnica, \$DSschedaTecnica)**

Metodo utilizzato per gestione allegati secondo le specifiche

**Parameters:**

Boolean	<i>\$new</i>	Se TRUE prodotto nuovo, FALSE prodotto esistente
int	<i>\$id_product</i>	Id prodotto
String	<i>\$FlschedaTecnica</i>	Nome file della scheda tecnica
String	<i>\$DSschedaTecnica</i>	Descrizione del file scheda tecnica

**insertAttribute ( \$id\_attribute\_group)**

Inserisce attributo nello shop

**Parameters:**

Int	<i>\$id_attribute_group</i>	Id del gruppo dell'attributo
-----	-----------------------------	------------------------------

**Returns:**

Int Id dell'attributo inserito

**insertAttributeGroup ()**

Se non presente attributo taglie, inserisco attributo e recupero id per aggiornare le altre tabelle

**Returns:**

Int Id attributo

**insertAttributeGroupLang ( \$id\_attribute\_group)**

Inserisco note attributo per ogni lingua dell'ecommerce

**Parameters:**

type	<i>\$id_attribute_group</i>	
------	-----------------------------	--

**insertAttributeGroupToShop ( \$id\_attribute\_group)**

Metodo utilizzato per inserire gli attributi negli shop definiti in Prestashop

**Parameters:**

Int	<i>\$id_attribute_group</i>	Id del gruppo
-----	-----------------------------	---------------

**insertAttributeShop ( \$id\_attribute)**

Collego valore attributo al negozio

**Parameters:**

Int	<i>\$id_attribute</i>	Id dell'attributo
-----	-----------------------	-------------------

**insertAttributeToProduct ( \$id\_product\_parent, \$data)**

Inserisco impatto valore attributo sul prodotto

**Parameters:**

Int	<i>\$id_product_parent</i>	Id del prodotto genitore
Array	<i>\$data</i>	Dati del prodotto da inserire dal tracciato dati ERP

**Returns:**

Int Id del prodotto Inserito / aggiornato

**insertCategoryInAllTree ( \$idProduct, \$id\_category\_default)**

Metodo per inserire il prodotto in tutte le categorie e sottocategorie

**Parameters:**

Int	<i>\$idProduct</i>	Id del prodotto da inserire nelle sotto categorie
Int	<i>\$id_category_default</i>	Id della categoria genitore

**insertCategoryProduct ( \$idProduct, \$idCategory)**

Metodo per inserire il prodotto in ogni categoria di appartenenza

**Parameters:**

Int	<i>\$idProduct</i>	Id del prodotto
-----	--------------------	-----------------



Int	<i>\$idCategory</i>	Id della categoria
-----	---------------------	--------------------

### **insertLangAttribute ( *\$value*, *\$id\_attribute* )**

Inserisco valore attributo per ogni lingua

#### **Parameters:**

String	<i>\$value</i>	Nome dell'attributo da inserire
Int	<i>\$id_attribute</i>	ID attributo da inserire

### **insertProduct ( *\$arrPrd*, *\$code\_product*, *\$has\_combination* )**

Metodo per inserire un prodotto all'interno del database di Prestashop

#### **Parameters:**

Array	<i>\$arrPrd</i>	Array dei dati del prodotto passati dal tracciato
String	<i>\$code_product</i>	Codice del prodotto
Boolean	<i>\$has_combination</i>	Flag che identifica se il prodotto ha combinazioni oppure no

#### **Returns:**

Int Id del prodotto appena creato

### **insertProductCombination ( *\$id\_attribute*, *\$id\_product\_attribute* )**

Inserisco combinazione valore attributo

#### **Parameters:**

int	<i>\$id_attribute</i>	Id attributo
int	<i>\$id_product_attribute</i>	Id del prodotto

### **insertProductImage2 ( *\$img\_1*, *\$position*, *\$id\_product* )**

Metodo per inserire immagine al prodotto

#### **Parameters:**

String	<i>\$img_1</i>	Nome dell'immagine del file
int	<i>\$position</i>	position immagine
int	<i>\$id_product</i>	Id del prodotto

### **insertProductLang ( *\$arrPrd*, *\$idProduct*, *\$lang* )**

Metodo utilizzato per inserire le etichette per i prodotti

#### **Parameters:**

Array	<i>\$arrPrd</i>	Array dei prodotti passati dal tracciato ERP
Id	<i>\$idProduct</i>	Id del prodotto inserito in ps_product
Array	<i>\$lang</i>	Array associativo contenente le localizzazioni del prodotto

### **insertProductShop ( *\$arrPrd*, *\$idProduct* )**

Metodo per inserire il prodotto all'interno degli shop configurati in Prestashop

#### **Parameters:**

Array	<i>\$arrPrd</i>	Array prodotto dei dati del tracciato ERP
Int	<i>\$idProduct</i>	Id del prodotto

### **insertProduttori ( )**

Metodo utilizzato per inserire i produttori ricavati dal tracciato dei prodotti di ERP

**insertRelatedProduct ( \$data)**

**insertStockAttrProduct ( \$id\_product\_attribute, \$data, \$id\_product)**

Metodo per inserire attributo del prodotto nello stock dello Shop

**Parameters:**

Int	<i>\$id_product_attribute</i>	Id attributo nel prodotto
Array	<i>\$data</i>	Array dei dati del prodotto dal tracciato ERP
Int	<i>\$id_product</i>	Id del prodotto

**insertStockProduct ( \$arrPrd, \$idProduct)**

Metodo utilizzato per inserire il prodotto all'interno dello stock del negozio

**Parameters:**

Array	<i>\$arrPrd</i>	Array con i dati del prodotto fornito dal tracciato ERP
int	<i>\$idProduct</i>	Id del prodotto

**insertValueAttributeShio ( \$id\_product\_attribute, \$data, \$id\_product)**

Inserisco valore attributo nel negozio

**Parameters:**

Int	<i>\$id_product_attribute</i>	Id dell'attributo del prodotto
Array	<i>\$data</i>	Dati del prodotto dal tracciato ERP
Int	<i>\$id_product</i>	Id del prodotto

**newAttachment ( \$id\_product, \$file\_name, \$description)**

Aggiungo un nuovo allegato al prodotto ed elimino eventuali collegamenti precedenti

**Parameters:**

int	<i>\$id_product</i>	Id del prodotto
String	<i>\$file_name</i>	Nome del file allegato
String	<i>\$description</i>	Descrizione del file allegato

**reinitImageLang ( \$id\_image)**

Cancella da db la vecchia immagine del prodotto

**Parameters:**

int	<i>\$id_image</i>	ID immagine
-----	-------------------	-------------

**reinitImageShop ( \$id\_image)**

Cancella da db shop immagine del prodotto

**Parameters:**

int	<i>\$id_image</i>	Id immagine
-----	-------------------	-------------

**reinitProductImage ( \$id\_product)**

Metodo per cancellare immagini al prodotto associata

**Parameters:**

int	<i>\$id_product</i>	Id del prodotto
-----	---------------------	-----------------

**resetImages ( \$id\_product)**

Resetta le immagini del prodotto

**Parameters:**

int	<i>\$id_product</i>	Id del prodotto
-----	---------------------	-----------------

**setProductPriceZero ( \$id\_product)**

Forzo il prezzo a zero

**Parameters:**

Int	<i>\$id_product</i>	Id del prodotto
-----	---------------------	-----------------

**unlinkImageProduct ( \$img\_prd\_pathdefault)**

Cancello le immagini precedenti

**Parameters:**

String	<i>\$img_prd_pathdefault</i>	Cartella relativa dell'immagine
--------	------------------------------	---------------------------------

**updateDescriptionAttachment ( \$id\_product, \$description)**

Update SOLO descrizione file allegato

**Parameters:**

int	<i>\$id_product</i>	Id del prodotto
String	<i>\$description</i>	Descrizione file allegato da aggiornare

**updateManufacturInsideProduct ( \$prod\_code, \$id\_manufacturer)**

Metodo per associare prodotto al produttore

**Parameters:**

String	<i>\$prod_code</i>	Codice del prodotto
int	<i>\$id_manufacturer</i>	Id produttore

**updateNameAttachment ( \$id\_product, \$file\_name)**

Update SOLO nome file allegato

**Parameters:**

int	<i>\$id_product</i>	Id del prodotto
String	<i>\$file_name</i>	Nome file allegato

**updateProduct ( \$arrPrd, \$id\_product, \$code\_product, \$has\_combination)**

Metodo che aggiorna prodotto esistente

**Parameters:**

Array	<i>\$arrPrd</i>	Array contenente i dati del prodotto dal tracciato ERP
int	<i>\$id_product</i>	Id del prodotto
String	<i>\$code_product</i>	Codice del prodotto

**Returns:**

int Id del prodotto aggiornato

**updateProductLang ( \$arrPrd, \$idProduct, \$lang)**

Metodo che aggiorna etichette di un prodotto esistente

**Parameters:**

Array	<i>\$arrPrd</i>	Array dei dati del prodotto da aggiornare dal tracciato ERP
Int	<i>\$idProduct</i>	Id del prodotto
Array	<i>\$lang</i>	Array associativo delle lingue del prodotto

**updateProductShop ( *\$arrPrd*, *\$idProduct* )**

Metodo per aggiornare il prodotto all'interno degli shop configurati nell'ecommerce

**Parameters:**

Array	<i>\$arrPrd</i>	Array di dati del prodotto dal tracciato ERP
Int	<i>\$idProduct</i>	Id del prodotto

**updateStockProduct ( *\$arrPrd*, *\$idProduct* )**

Metodo per aggiornare il prodotto nello stock del negozio

**Parameters:**

Array	<i>\$arrPrd</i>	Array dei dati del prodotto dal tracciato ERP
Int	<i>\$idProduct</i>	Id del prodotto

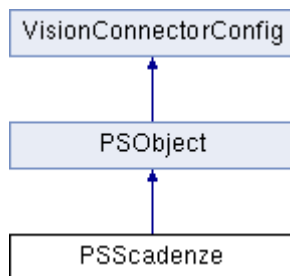
---

The documentation for this class was generated from the following file:

- classes/**PSPProduct.php**

## PSScadenze Class Reference

Inheritance diagram for PSScadenze:



### Public Member Functions

- `getScadenze ()`
- `getTipiScadenze ()`
- `getTipiEffetto ()`
- `insertScadenza ($data)`
- `insertTipoScadenza ($data)`
- `insertTipoEffettoScadenza ($data)`

### Additional Inherited Members

---

### Detailed Description

Description of **PSScadenze**

**Author:**

ALLINITSRLS

---

### Member Function Documentation

**getScadenze ()**

**getTipiEffetto ()**

**getTipiScadenze ()**

**insertScadenza ( *\$data*)**

**insertTipoEffettoScadenza ( *\$data*)**

**insertTipoScadenza ( *\$data*)**

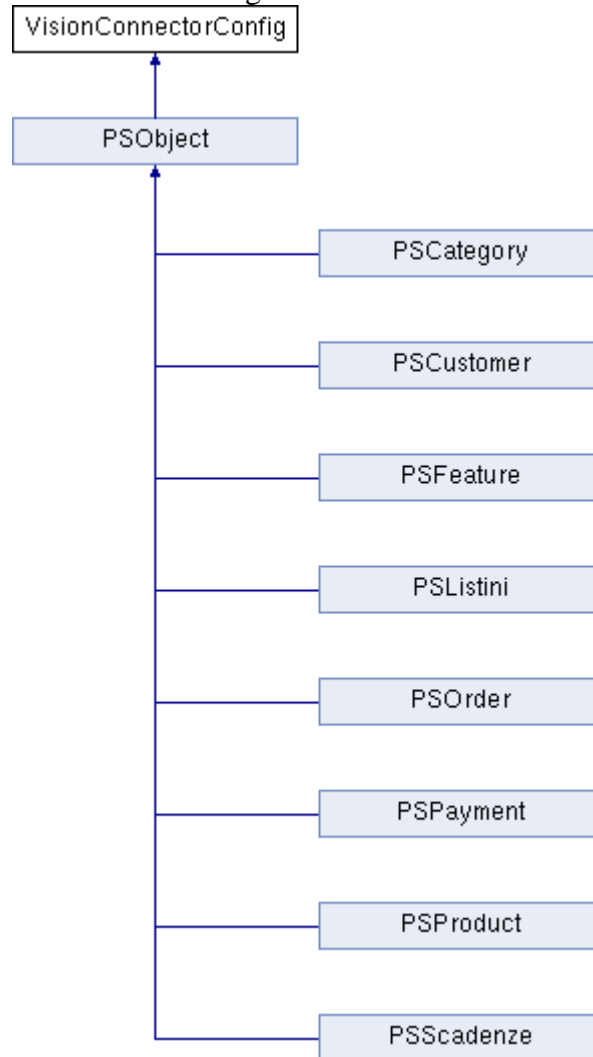
---

The documentation for this class was generated from the following file:

- `classes/PSScadenze.php`

## VisionConnectorConfig Class Reference

Inheritance diagram for VisionConnectorConfig:



### Data Fields

- `$ecommerce_lang_by_erp` = array("IT" => 1)
- `$ecomemrce_lang_by_ps` = array(1 => "IT")
- `$default_language_ecommerce` = "IT"

---

### Field Documentation

`$default_language_ecommerce` = "IT"

`$ecomemrce_lang_by_ps` = array(1 => "IT")

`$ecommerce_lang_by_erp` = array("IT" => 1)

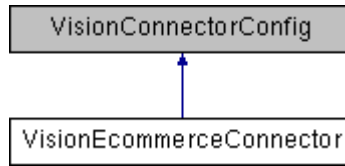
---

The documentation for this class was generated from the following file:

- `conf/config.connector.php`

## VisionEcommerceConnector Class Reference

Inheritance diagram for VisionEcommerceConnector:



### Public Member Functions

- `__construct ()`
- `getLastOrdineId ()`
- `getOrdini ()`
- `getCodAnaFromId ($id)`
- `unselectOrder ($id)`
- `getSingleOrdini ()`
- `getCodProdFromID ($id)`
- `getOrderDataFromID ($id)`
- `unselectSingleOrder ($id)`
- `getAnagrafERP ()`
- `getCodeClienti ()`
- `fillCodeClientiWithZero ($prefix, $id_cliente)`
- `getSingleAnagFromId ($id)`
- `getStateFromId ($id_state)`
- `unselectSingleAnag ($id, $code)`
- `createDoneFile ()`
- `getIndirizzi ()`
- `setAnagraficaSynchronized ($idAnagrafica)`

### Data Fields

- `$file_version_importer = "CONN_VER"`
- `$file_info = "TPD_INFO"`
- `$file_category = "TPD_PRODOTTI_CATEGORIE"`
- `$file_category_lang = "TPD_PRODOTTI_LINGUA_CATEGORIE"`
- `$file_product = "TPD_PRODOTTI"`
- `$file_product_lang = "TPD_PRODOTTI_LINGUA"`
- `$file_customer = "TSD_UTENTI"`
- `$file_anagrafiche = "TSD_ANAGRAFICHE"`
- `$file_listini_base = "TPD_LISTINI_BASE"`
- `$file_listini = "TPD_LISTINI"`
- `$last_ordine_id = 0`

---

## Constructor & Destructor Documentation

### `__construct ()`

Costruttore della classe PS2ERP per esportare i dati necessari da PRESTASHOP verso VISIONERP

---



## Member Function Documentation

### createDoneFile ()

Metodo per a creazione del file di fine operazione Prestasop TO VisionERP complete.done

### fillCodeClientiWithZero ( *\$prefix*, *\$id\_cliente*)

### getAnagrafERP ()

Metodo per il recupero delle anagrafiche presenti dentro Prestashop e la creazione del file FWCLIENTI.txt

### getCodAnaFromId ( *\$id*)

Metodo per recuperare il codice di un'anagrafica (identificativo di VisionERP) a partire dall'ID di Prestashop della tabella ps\_customer

#### Parameters:

int	<i>\$id</i>	Id customer Prestashop della tabella ps_customer
-----	-------------	--

#### Returns:

CdAnag Restituisce il codice dell'anagrafica univoco per VisionERP

### getCodeClienti ()

Metodo per il PREFISSO impostato da VisionERP, per identificare i clienti WEB.

Il valore viene passato nel tracciato nella fase di import da VisionERP a Prestashop nella tabella vsecommerce\_info con flag CODICECLI

#### Returns:

CODICECLI Codice Cliente da associare ai clienti che si registrano da Web

### getCodProdFromID ( *\$id*)

Metodo per restituire il codice del prodotto VisionERP dal Prestashop dalla tabella ps\_product

#### Parameters:

int	<i>\$id</i>	Id dell'ordine
-----	-------------	----------------

#### Returns:

Code Codice VisioERP del prodotto

### getIndirizzi ()

Metodo per a creazione del file di indirizzi (indirizzi fatturazione - spedizione)

### getLastOrdineld ()

### getOrderDataFromID ( *\$id*)

Metodo per recuperare la data di inserimento dell'ordine in Prestashop

#### Parameters:

int	<i>\$id</i>	ID dell'ordine PRestashop
-----	-------------	---------------------------

#### Returns:

date\_add Data e Ora inserimento ordine in Prestashop

### getOrdini ()

Metodo utilizzato per il recupero degli ordini su Prestashop Utilizza il flag `erp_diff` della tabella `ps_orders`.  
Tutti gli ordini che hanno `erp_diff = 0` vengono esportati nel tracciato del file `FWDOCTES.txt`

### getSingleAnagFromId ( \$id)

Metodo per recuperare intera Anagrafica (indirizzo `ps_address`) da Prestashop

#### Parameters:

int	<i>\$id</i>	ID del customer Prestashop
-----	-------------	----------------------------

#### Returns:

ANAGRAFICA Record della tabella `ps_address` Prestashop

### getSingleOrdini ()

Metodo per recuperare i dettagli degli ordini processati

### getStateFromId ( \$id\_state)

Metodo per recuperare la stringa `ISO_CODE` della provincia del customer Prestashop

#### Parameters:

int	<i>\$id_state</i>	Identificativo della provincia associata al customer
-----	-------------------	--

#### Returns:

`ISO_CODE` Sigla della provincia del customer Prestashop

### setAnagraficaSynchronized ( \$idAnagrafica)

### unselectOrder ( \$id)

Metodo per deselzionare ordine appena esportato dalla lista.

Serve a non far restituire due volte lo stesso valore a VisionERP

#### Parameters:

int	<i>\$id</i>	ID dell'ordine di Prestashop della tabella <code>ps_orders</code>
-----	-------------	---

### unselectSingleAnag ( \$id, \$code)

Metodo per deselezionare la singola anagrafica esportata precedentemente da Prestashop verso VisionERP

Nel momento dell'unselect viene anche configurato il CdAnag WEB

#### Parameters:

int	<i>\$id</i>	ID del customer Pestashop
string	<i>\$code</i>	CdAnag del customer

### unselectSingleOrder ( \$id)

Metodo per deselzionare dal successivo export verso VisionERP il dettaglio dell'ordine già esportato

#### Parameters:

int	<i>\$id</i>	Id ordine di Prestashop
-----	-------------	-------------------------

## Field Documentation

`$file_anagrafiche = "TSD_ANAGRAFICHE"`

`$file_category = "TPD_PRODOTTI_CATEGORIE"`

`$file_category_lang = "TPD_PRODOTTI_LINGUA_CATEGORIE"`

`$file_customer = "TSD_UTENTI"`

`$file_info = "TPD_INFO"`

`$file_listini = "TPD_LISTINI"`

`$file_listini_base = "TPD_LISTINI_BASE"`

`$file_product = "TPD_PRODOTTI"`

`$file_product_lang = "TPD_PRODOTTI_LINGUA"`

`$file_version_importer = "CONN_VER"`

`$last_ordine_id = 0`

---

The documentation for this class was generated from the following file:

- `ps2erp.php`

# File Documentation

## classes/PSCategory.php File Reference

### Data Structures

- class **PSCategory**

### Namespaces

- **PSCategory**
- **AIPSCollector**

### Variables

- const **FILE\_CATEGORY** 'TPD\_PRODOTTI\_CATEGORIE'
- const **FILE\_CATEGORY\_LANG** 'TPD\_PRODOTTI\_LINGUA\_CATEGORIE'
- const **FOLDER\_CATEGORY\_IMG** PATH\_ABSOLUTE\_AICONNECTOR . "/img/c/"
- const **TOPMENU\_ENABLECAT** "1"

## classes/PSCustomer.php File Reference

### Data Structures

- class **PSCustomer**

### Namespaces

- **AIPSCollector**

### Variables

- const **FILE\_CUSTOMER** "TSD\_UTENTI"
- const **FILE\_ANAGRAFICHE** "TSD\_ANAGRAFICHE"

## classes/PSFeature.php File Reference

### Data Structures

- class **PSFeature**

### Namespaces

- **AIPSCollector**

### Variables

- const **URL\_WGET\_RELOAD\_ATTRIBUTE**  
"http://devps.all-init.it/modules/blocklayered/blocklayered-attribute-indexer.php?token=f550d3b96e"
- const **URL\_WGET\_RELOAD\_ATTRIBUTE\_URL**  
"http://devps.all-init.it/modules/blocklayered/blocklayered-url-indexer.php?token=f550d3b96e&truncate=1"

## classes/PSListini.php File Reference

### Data Structures

- class **PSListini**

### Namespaces

- **AIPSCollector**

### Variables

- const **FILE\_PRODUCT** "TPD\_PRODOTTI"
- const **FILE\_LISTINI\_BASE** "TPD\_LISTINI\_BASE"
- const **FILE\_LISTINI** "TPD\_LISTINI"
- const **FILE\_SCONTI\_QT** "TSD\_SCONTI\_SCAGLIONE"
- const **LIST\_BASE\_NUMBER** 2
- const **PROPAPAGAZIONE\_LISTINI\_CLIENTI** 0

## classes/PObject.php File Reference

### Data Structures

- class **PObject**

### Namespaces

- **AIPSCollector**

### Variables

- `$FILE_LOG_NAME = "log_import_" . date("YmdHis") . ".txt"`
- `const FILE_VERSION_ERP "CONN_VER"`
- `const FILE_INFO "TPD_INFO"`



## classes/PSOrder.php File Reference

### Data Structures

- class **PSOrder**

### Namespaces

- **AIPSCollector**

### Variables

- const **FOLDER\_PDF** `PATH_ABSOLUTE_AICollector . "/img/p/pdf/"`
- const **FOLDER\_TMP\_PDF** `PATH_ABSOLUTE_AICollector . "/img/pdftmp/"`

## **classes/PSPayment.php File Reference**

### **Data Structures**

- class **PSPayment**

### **Namespaces**

- **AIPSCollector**

## classes/PSPProduct.php File Reference

### Data Structures

- class **PSPProduct**

### Namespaces

- **AIPSCollector**

### Variables

- const **FILE\_PRODUCT\_LINGUA** "TPD\_PRODOTTI\_LINGUA"
- const **ENABLE\_TAGLIE** 0
- const **DELIMITATORE\_TAGLIE** "èéà"
- const **FILL\_PARENT\_CATEGORY** 1
- const **FOLDER\_ATTACHMENT** PATH\_ABSOLUTE\_AICONNECTOR . "/download"

## **classes/PSScadenze.php File Reference**

### **Data Structures**

- class **PSScadenze**

### **Namespaces**

- **AIPSConnector**

## **cleanPSCache.php File Reference**

## conf/config.connector.php File Reference

### Data Structures

- class **VisionConnectorConfig**

### Namespaces

- **AIPSCollector**

### Variables

- const **DB\_HOST** "localhost"
- const **DB\_USER** 'gfranchitto'
- const **DB\_PASS** 'crekepispi'
- const **DB\_NAME** "devps"
- const **PATH\_ABSOLUTE\_AICONNECTOR** "/home/gfranchitto/domains/devps.all-init.it/public\_html"
- const **FOLDER\_UNZIP** PATH\_ABSOLUTE\_AICONNECTOR . "/visionImport/tmp\_data/"
- const **FOLDER\_PS\_IMAGE** PATH\_ABSOLUTE\_AICONNECTOR . "/img/p/"
- const **FOLDER\_TMP\_IMG** PATH\_ABSOLUTE\_AICONNECTOR . "/img/tmp/"
- const **FOLDER\_IMPORT\_ROOT** PATH\_ABSOLUTE\_AICONNECTOR . "/visionImport/"
- const **FILE\_ERP\_DATA** "d\_oi\_b2b.zip"
- const **FILE\_ERP\_IMG** "oi\_b2b\_files.zip"
- const **WIN\_CONNECTOR\_VER** "04"
- const **SALT\_PASSWORD** "3vLbIAfx0jM8WBkFuaMRmtwkgdJktPgHFU9G0vBka783cOtiFMb10KW"
- const **ADMIN\_FOLDER** "admin4675"
- const **\_PS\_ADMIN\_DIR** PATH\_ABSOLUTE\_AICONNECTOR . '/' . ADMIN\_FOLDER
- const **EMAIL\_LOG\_TO** "gfranchitto@all-init.it"
- const **EMAIL\_LOG\_TO\_2** "gfranchitto2@all-init.it"
- const **EMAIL\_LOG\_FROM** "notifiche@all-init.it"
- const **EMAIL\_LOG\_SUBJECT** "[DEVPS - REPORT] Export Product to Ecommerce"
- const **LANG\_NUMBER** 1
- const **SHOP\_NAME** 'DEVPS Ecommerce'
- const **SHOP\_URL** "http://devps.all-init.it/"
- const **SHOP\_LOGO** "http://devps.all-init.it/img/logo.jpg"
- const **SUBJ\_ACTIVATEEMAIL** "[DEVPS ALLINIT] - Attivazione account"
- const **FROM\_ACTIVATEEMAIL** "devps@all-init.it"

## endpnt\_export.php File Reference

### Functions

- `sendMailAlert ()`

### Variables

- `$psOutput = null`
- `$alreadyExecuted = FALSE`
- `foreach($psOutput as $row) if($alreadyExecuted) else`

---

### Function Documentation

`sendMailAlert ()`

---

### Variable Documentation

`$alreadyExecuted = FALSE`

`$psOutput = null`

`foreach ($psOutput as $row) if ($alreadyExecuted) else`

```
Initial value: {  
    shell_exec('php erp2ps.php ' . md5(ADMIN_FOLDER) . ' > /dev/null 2>/dev/null &')
```

## **endpnt\_import.php File Reference**



## erp2ps.php File Reference

### Variables

- `const VISION_IMPORT_VERSION "2.0.0"`
  - `$ps_category = new \AIPSCollector\PSCategory()`
  - `if(file_exists(FOLDER_IMPORT_ROOT.FILE_ERP_DATA)) else`
  - `$ps_product = new AIPSCollector\PSProduct()`
  - `if($ps_product->checkIfThereAreImages()) $ps_customer = new \AIPSCollector\PSCustomer()`
  - `$ps_listini = new AIPSCollector\PSListini()`
  - `$ps_feature = new \AIPSCollector\PSFeature()`
  - `$ps_payment = new \AIPSCollector\PSPayment()`
  - `$ps_orders = new AIPSCollector\PSOrder()`
  - `$ps_scadenze = new AIPSCollector\PSScadenze()`
- 

### Variable Documentation

`$ps_category = new \AIPSCollector\PSCategory()`

`if ($ps_product->checkIfThereAreImages()) $ps_customer = new \AIPSCollector\PSCustomer()`

`$ps_feature = new \AIPSCollector\PSFeature()`

`$ps_listini = new AIPSCollector\PSListini()`

`$ps_orders = new AIPSCollector\PSOrder()`

`$ps_payment = new \AIPSCollector\PSPayment()`

`$ps_product = new AIPSCollector\PSProduct()`

`$ps_scadenze = new AIPSCollector\PSScadenze()`

`if ($ps_category->checkVersionImporter()) else`

```
Initial value:{
    $ps_category->fileLog("#2: ERRORE " . FILE_ERP_DATA . " NON PRESENTE")
}
```

`const VISION_IMPORT_VERSION "2.0.0"`

Definizione della versione del Connector PHP

## lib/class.phpmailer.php File Reference

### Data Structures

- class **PHPMailer**
- class **phpmailerException**

### Namespaces

- **PHPMailer**

### Variables

- `if(version_compare(PHP_VERSION, '5.0.0', '<')) exit("Sorry this version of PHPMailer will only run on PHP version or greater n`

---

### Variable Documentation

`if (version_compare(PHP_VERSION, '5.0.0', '<')) exit("Sorry this version of PHPMailer will only run on PHP version or greater n`

## lib/ERP.class.php File Reference

### Data Structures

- class **ERP\_TSD\_ANAGRAFICHE**
- class **ERP\_TSD\_UTENTI**
- class **ERP\_TPD\_LISTINI\_BASE**
- class **ERP\_TPD\_PRODOTTI**
- class **PS\_specific\_price**
- class **ERP\_TSD\_SCONTI\_SCAGLIONE**
- class **ERP\_TPD\_LISTINI**

## **lib/mysql.class.php File Reference**

### **Data Structures**

- class MySQL

### **Namespaces**

- AIPSCollector

