

OWL-PHP Reference Manual
010

Generated by Doxygen 1.5.4

Mon Sep 8 14:26:30 2008

Contents

1	OWL-PHP Module Index	1
1.1	OWL-PHP Modules	1
2	OWL-PHP Hierarchical Index	3
2.1	OWL-PHP Class Hierarchy	3
3	OWL-PHP Class Index	5
3.1	OWL-PHP Class List	5
4	OWL-PHP File Index	7
4.1	OWL-PHP File List	7
5	OWL-PHP Module Documentation	9
5.1	Presentation modules	9
5.2	Business Object modules	10
5.3	Storage Object modules	11
5.4	Library (codes, messages files etc.)	12
6	OWL-PHP Class Documentation	13
6.1	_OWL Class Reference	13
6.2	ConfigHandler Class Reference	16
6.3	DataHandler Class Reference	18
6.4	DbHandler Class Reference	24
6.5	DOMElement Class Reference	31
6.6	FileHandler Class Reference	33
6.7	Form Class Reference	35
6.8	FormElement Class Reference	37
6.9	FormHandler Class Reference	39
6.10	LogHandler Class Reference	44
6.11	OWL Class Reference	48

6.12	OWLException Class Reference	50
6.13	OWLExceptionHandler Class Reference	52
6.14	Register Class Reference	53
6.15	Session Class Reference	56
6.16	SessionHandler Class Reference	61
6.17	StatusHandler Class Reference	66
6.18	User Class Reference	69
6.19	UserHandler Class Reference	74
7	OWL-PHP File Documentation	79
7.1	/home/oscar/work/eclipse/owl-php/src/config.php File Reference	79
7.2	/home/oscar/work/eclipse/owl-php/src/index.php File Reference	81
7.3	/home/oscar/work/eclipse/owl-php/src/kernel/bo/class.form.php File Reference .	82
7.4	/home/oscar/work/eclipse/owl-php/src/kernel/bo/class.session.php File Reference	83
7.5	/home/oscar/work/eclipse/owl-php/src/kernel/bo/class.user.php File Reference . .	84
7.6	/home/oscar/work/eclipse/owl-php/src/kernel/class._OWL.php File Reference . .	85
7.7	/home/oscar/work/eclipse/owl-php/src/kernel/so/class.confighandler.php File Reference	86
7.8	/home/oscar/work/eclipse/owl-php/src/kernel/so/class.datahandler.php File Reference	87
7.9	/home/oscar/work/eclipse/owl-php/src/kernel/so/class.dbhandler.php File Reference	90
7.10	/home/oscar/work/eclipse/owl-php/src/kernel/so/class.exceptionhandler.php File Reference	92
7.11	/home/oscar/work/eclipse/owl-php/src/kernel/so/class.filehandler.php File Reference	93
7.12	/home/oscar/work/eclipse/owl-php/src/kernel/so/class.formhandler.php File Reference	94
7.13	/home/oscar/work/eclipse/owl-php/src/kernel/so/class.loghandler.php File Reference	95
7.14	/home/oscar/work/eclipse/owl-php/src/kernel/so/class.register.php File Reference	96
7.15	/home/oscar/work/eclipse/owl-php/src/kernel/so/class.sessionhandler.php File Reference	98
7.16	/home/oscar/work/eclipse/owl-php/src/kernel/so/class.statushandler.php File Reference	99
7.17	/home/oscar/work/eclipse/owl-php/src/kernel/so/class.userhandler.php File Reference	100
7.18	/home/oscar/work/eclipse/owl-php/src/kernel/ui/class.domelement.php File Reference	101
7.19	/home/oscar/work/eclipse/owl-php/src/kernel/ui/class.formelement.php File Reference	102
7.20	/home/oscar/work/eclipse/owl-php/src/lib/owl.messages.en-uk.php File Reference	103
7.21	/home/oscar/work/eclipse/owl-php/src/lib/owl.messages.php File Reference	104
7.22	/home/oscar/work/eclipse/owl-php/src/lib/owl.severitycodes.php File Reference .	105
7.23	/home/oscar/work/eclipse/owl-php/src/OWLloader.php File Reference	107
7.24	/home/oscar/work/eclipse/owl-php/src/OWLrundown.php File Reference	109

Chapter 1

OWL-PHP Module Index

1.1 OWL-PHP Modules

Here is a list of all modules:

Presentation modules	9
Business Object modules	10
Storage Object modules	11
Library (codes, messages files etc.)	12

Chapter 2

OWL-PHP Hierarchical Index

2.1 OWL-PHP Class Hierarchy

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

_OWL	13
DataHandler	18
FormHandler	39
DbHandler	24
DOMElement	31
Form	35
FormElement	37
LogHandler	44
OWL	48
SessionHandler	61
Session	56
UserHandler	74
User	69
ConfigHandler	16
FileHandler	33
OWLException	50
OWLExceptionHandler	52
Register	53
StatusHandler	66

Chapter 3

OWL-PHP Class Index

3.1 OWL-PHP Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

_OWL	13
ConfigHandler (Configuration handler)	16
DataHandler (The OWL Data object)	18
DbHandler (Database handler)	24
DOMElement (DOM Element base class)	31
FileHandler (File handler)	33
Form (Form Element class)	35
FormElement (Form element)	37
FormHandler (Formhandler)	39
LogHandler (Log handler)	44
OWL	48
OWLException (Exception handler)	50
OWLExceptionHandler (Default exception handler)	52
Register	53
Session (OWL-PHP session object)	56
SessionHandler (PHP session object)	61
StatusHandler (Status object)	66
User (OWL-PHP session object)	69
UserHandler (User object)	74

Chapter 4

OWL-PHP File Index

4.1 OWL-PHP File List

Here is a list of all files with brief descriptions:

/home/oscar/work/eclipse/owl-php/src/ config.php	79
/home/oscar/work/eclipse/owl-php/src/ index.php	81
/home/oscar/work/eclipse/owl-php/src/ OWLloader.php	107
/home/oscar/work/eclipse/owl-php/src/ OWLrundown.php	109
/home/oscar/work/eclipse/owl-php/src/kernel/ class._OWL.php	85
/home/oscar/work/eclipse/owl-php/src/kernel/bo/ class.form.php	82
/home/oscar/work/eclipse/owl-php/src/kernel/bo/ class.session.php	83
/home/oscar/work/eclipse/owl-php/src/kernel/bo/ class.user.php	84
/home/oscar/work/eclipse/owl-php/src/kernel/so/ class.confighandler.php	86
/home/oscar/work/eclipse/owl-php/src/kernel/so/ class.datahandler.php	87
/home/oscar/work/eclipse/owl-php/src/kernel/so/ class.dbhandler.php	90
/home/oscar/work/eclipse/owl-php/src/kernel/so/ class.exceptionhandler.php	92
/home/oscar/work/eclipse/owl-php/src/kernel/so/ class.filehandler.php	93
/home/oscar/work/eclipse/owl-php/src/kernel/so/ class.formhandler.php	94
/home/oscar/work/eclipse/owl-php/src/kernel/so/ class.loghandler.php	95
/home/oscar/work/eclipse/owl-php/src/kernel/so/ class.register.php	96
/home/oscar/work/eclipse/owl-php/src/kernel/so/ class.sessionhandler.php	98
/home/oscar/work/eclipse/owl-php/src/kernel/so/ class.statushandler.php	99
/home/oscar/work/eclipse/owl-php/src/kernel/so/ class.userhandler.php	100
/home/oscar/work/eclipse/owl-php/src/kernel/ui/ class.domelement.php	101
/home/oscar/work/eclipse/owl-php/src/kernel/ui/ class.formelement.php	102
/home/oscar/work/eclipse/owl-php/src/lib/ owl.messages.en-uk.php	103
/home/oscar/work/eclipse/owl-php/src/lib/ owl.messages.php	104
/home/oscar/work/eclipse/owl-php/src/lib/ owl.severitycodes.php	105

Chapter 5

OWL-PHP Module Documentation

5.1 Presentation modules

Classes

- class [DOMElement](#)
DOM Element base class.
- class [FormElement](#)
Form element.

5.2 Business Object modules

Classes

- class [Form](#)
Form Element class.
- class [Session](#)
the OWL-PHP session object
- class [User](#)
the OWL-PHP session object

5.3 Storage Object modules

Classes

- class [ConfigHandler](#)
Configuration handler.
- class [DataHandler](#)
The [OWL](#) Data object.
- class [DbHandler](#)
Database handler.
- class [OWLException](#)
Exception handler.
- class [OWLExceptionHandler](#)
Default exception handler.
- class [FileHandler](#)
File handler.
- class [FormHandler](#)
Formhandler.
- class [LogHandler](#)
Log handler.
- class [Register](#)
- class [SessionHandler](#)
the PHP session object
- class [StatusHandler](#)
Status object.
- class [UserHandler](#)
the user object

5.4 Library (codes, messages files etc.)

Files

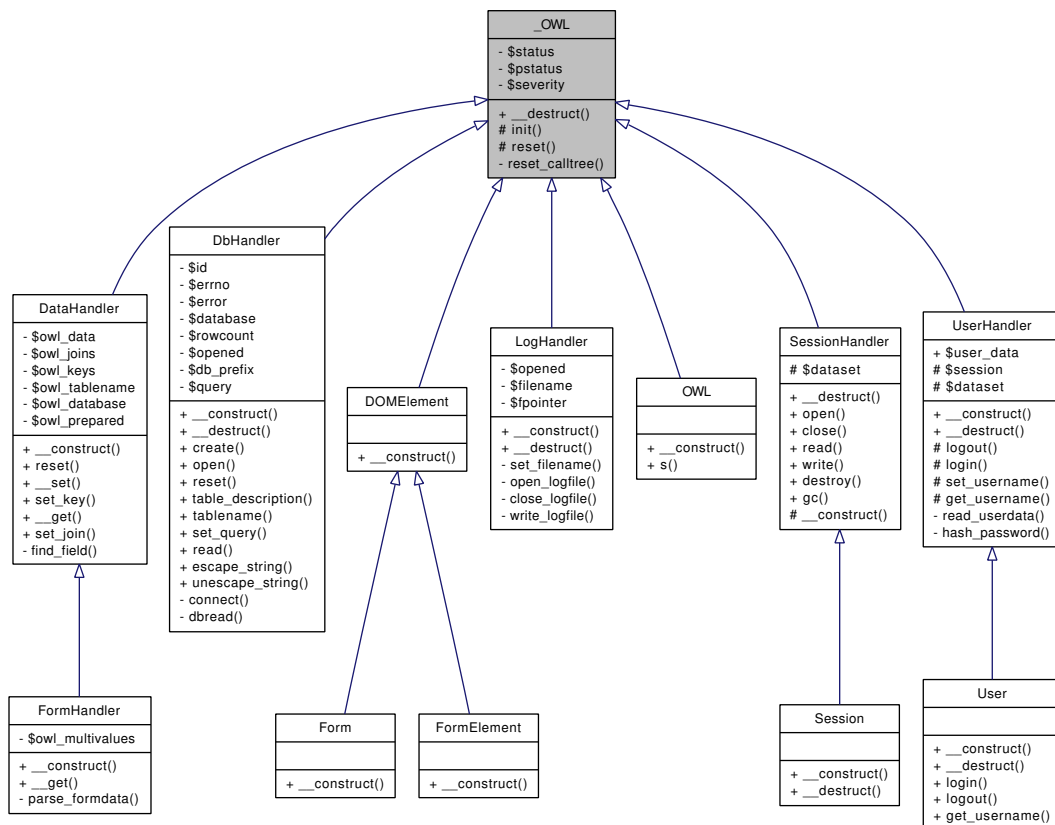
- file [owl.messages.en-uk.php](#)
- file [owl.messages.php](#)
- file [owl.severitycodes.php](#)
- file [OWLloader.php](#)
- file [OWLrundown.php](#)

Chapter 6

OWL-PHP Class Documentation

6.1 `_OWL` Class Reference

Inheritance diagram for `_OWL`:



Public Member Functions

- [__destruct \(\)](#)

Protected Member Functions

- [init \(\)](#)
- [reset \(\)](#)

Private Member Functions

- [reset_calltree \(\\$depth=0\)](#)

Private Attributes

- [\\$status](#)
- [\\$pstatus](#)
- [\\$severity](#)

6.1.1 Detailed Description

This is the main class for all [OWL](#) objects. It contains some methods that have to be available to all objects. Some of them can be reimplemented.

Author:

Oscar van Eijk, Oveas Functionality Provider

Version:

May 15, 2007 – O van Eijk – initial version

6.1.2 Constructor & Destructor Documentation

6.1.2.1 [__OWL::__destruct \(\)](#)

Default class destructor; has to exist but can (should?) be reimplemented

Reimplemented in [Session](#), [User](#), [DbHandler](#), [LogHandler](#), [SessionHandler](#), and [UserHandler](#).

6.1.3 Member Function Documentation

6.1.3.1 [__OWL::init \(\)](#) [protected]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this **might** indicate something went wrong.

6.1.3.2 `__OWL::reset ()` [protected]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#).

6.1.3.3 `__OWL::reset_calltree ($ depth = 0)` [final, private]

Reset the status in the complete calltree

Parameters:

← *\$depth* Keep track of the depth in recursive calls. Should be empty in the first call.

6.1.4 Member Data Documentation

6.1.4.1 `__OWL::$status` [private]

Current object status

6.1.4.2 `__OWL::$pstatus` [private]

Pointer to the object which holds the last nonsuccessfull (\geq `OWL_WARNING`) status

6.1.4.3 `__OWL::$severity` [private]

Severity level of the current object status

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

- `/home/oscar/work/eclipse/owl-php/src/kernel/class._OWL.php`

6.2 ConfigHandler Class Reference

Configuration handler.

Public Member Functions

- [read_config](#) (\$file= ")
- [get](#) (\$item)

Private Member Functions

- [convert](#) (\$val)

6.2.1 Detailed Description

Configuration handler.

This abstract class reads configuration from file or database, and fills and reads the global data-structure with config items

Author:

Oscar van Eijk, Oveas Functionality Provider

Version:

Aug 20, 2008 – O van Eijk – initial version

6.2.2 Member Function Documentation

6.2.2.1 ConfigHandler::read_config (\$ *file* = ")

Find out what the filename of the logfile should be

6.2.2.2 ConfigHandler::convert (\$ *val*) [private]

Convert values in character string format to a known value

Parameters:

← *\$val* The value as read from the config file

Returns:

Value in the desired format (or as is if nothing set)

6.2.2.3 ConfigHandler::get (\$ *item*)

Return a configuration value. Note! In order to use hidden values properly, this is the **ONLY** way configuration values should be retrieved!

Parameters:

- ← *\$item* The configuration item in the same format as it appears in the configuration file (e.g. 'group|subject|item')

Returns:

Corresponding value or null when nothing was found

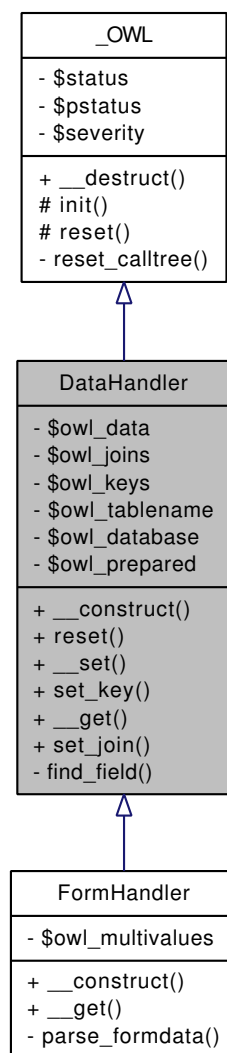
The documentation for this class was generated from the following file:

- /home/oscar/work/eclipse/owl-php/src/kernel/so/[class.confighandler.php](#)

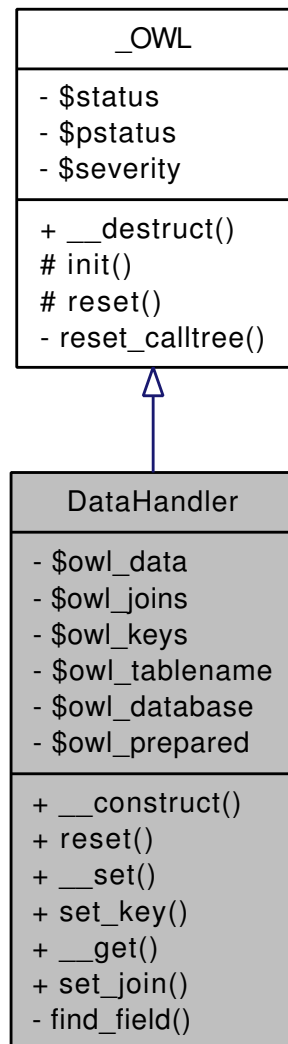
6.3 DataHandler Class Reference

The [OWL](#) Data object.

Inheritance diagram for DataHandler:



Collaboration diagram for DataHandler:



Public Member Functions

- `__construct` (\$dblink=null, \$tablename= "")
- `reset` (\$level=DATA_RESET_PREPARE)
- `__set` (\$variable, \$value)
- `set_key` (\$variable)
- `__get` (\$variable)
- `set_join` (\$lvalue, \$rvalue, \$linktype= '=')

Protected Member Functions

- `init` ()
- `reset` ()

Private Member Functions

- [find_field](#) (\$fld, &\$expanded)

Private Attributes

- [\\$owl_data](#)
- [\\$owl_joins](#)
- [\\$owl_keys](#)
- [\\$owl_tablename](#)
- [\\$owl_database](#)
- [\\$owl_prepared](#)

6.3.1 Detailed Description

The [OWL](#) Data object.

This class contains DB datasets

Author:

Oscar van Eijk, Oveas Functionality Provider

Version:

Aug 4, 2008 – O van Eijk – initial version

6.3.2 Constructor & Destructor Documentation

6.3.2.1 DataHandler::__construct (\$ *dblink* = null, \$ *tablename* = ”)

Class constructor

Parameters:

- ← *\$dblink* Database object
- ← *\$tablename* Default table name for this dataset

6.3.3 Member Function Documentation

6.3.3.1 DataHandler::reset (\$ *level* = DATA_RESET_PREPARE)

Reset the object

Parameters:

- ← *\$level* The reset level, can be any of the following:

- DATA_RESET_STATUS; Reset object status only
- DATA_RESET_PREPARE (default); Reset status and a prepared query
- DATA_RESET_META; Reset status and query, remove all locks and joins
- DATA_RESET_FULL; Reset status and query, remove all locks, joins and data

6.3.3.2 DataHandler::__set (\$ variable, \$ value)

Define or override a variable in the data array

Parameters:

- ← *\$variable* The name of the variable that should be set
- ← *\$value* Value to set the variable to. If this is an array, the second field in the array has to be the tablename where the fieldname is found. The value itself van NEVER be an array!

6.3.3.3 DataHandler::set_key (\$ variable)

Lock variables for update by adding them to an array. Fields in this array will not be overwritten on updates, but used in WHERE clauses.

Parameters:

- ← *\$variable* Variable name to lock, optionally as an array (table, field)

Returns:

Severity level

6.3.3.4 DataHandler::find_field (\$ fld, &\$ expanded) [private]

Try to expand a field to a fully qualified 'table#field' name

Parameters:

- ← *\$fld* The fieldname that has to be expanded
- *\$expanded* An array with all matching fully qualified fieldnames.

Returns:

The number of matches

6.3.3.5 DataHandler::__get (\$ variable)

Retrieve a value from the data array. The variable name can be a fully qualified table/fieldname (format "table#field"), or only a field name, in which case it has to be unique. If the fieldname cannot be found directly, the array is scanned to find a matching field. If more matches are found, the object status is set to DATA_AMBFIELD.

Parameters:

← *\$variable* The name of the variable that should be retrieved

Returns:

The value, or NULL when the value was not found or ambiguous.

Reimplemented in [FormHandler](#).

6.3.3.6 DataHandler::set_join (\$ lvalue, \$ rvalue, \$ linktype = '=')

Define a link between 2 fields that will be recognized when the database query is built.

Parameters:

← *\$lvalue* Left value as array(table, field)

← *\$rvalue* Right value as array(table, field)

← *\$linktype* How are the fields linked. Can be any binary operator as recognized by SQL.

Returns:

Severity level

6.3.3.7 __OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

6.3.3.8 __OWL::reset () [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#).

6.3.4 Member Data Documentation

6.3.4.1 DataHandler::\$owl_data [private]

Indexed array holding all data values.

6.3.4.2 DataHandler::\$owl_joins [private]

2D Array holding all relationships between the data.

6.3.4.3 DataHandler::\$owl_keys [private]

Array with variable names that are used in WHERE clauses on updates

6.3.4.4 DataHandler::\$owl_tablename [private]

All variable names are expected to be fields in a database as well. If a table name is not given, the default table name will be used. This is useful for datasets that come from only one database table. For datasets that are not read from or written to a database, the tablename can be null.

6.3.4.5 DataHandler::\$owl_database [private]

An optional link to a database object. This has to be specified if the data needs to be written to or read from a database.

6.3.4.6 DataHandler::\$owl_prepared [private]

Boolean that indicates if a query has been prepared

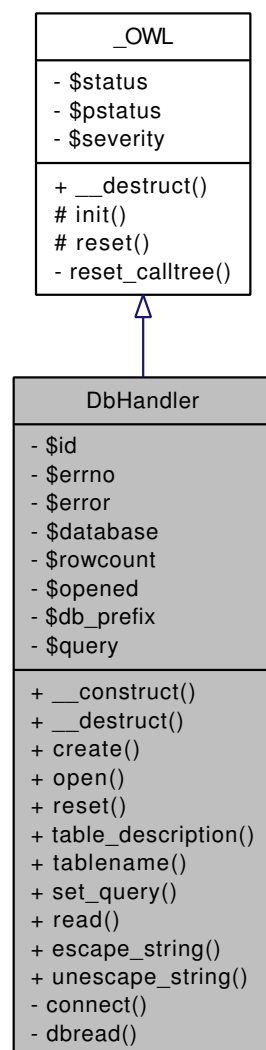
The documentation for this class was generated from the following file:

- [/home/oscar/work/eclipse/owl-php/src/kernel/so/class.datahandler.php](#)

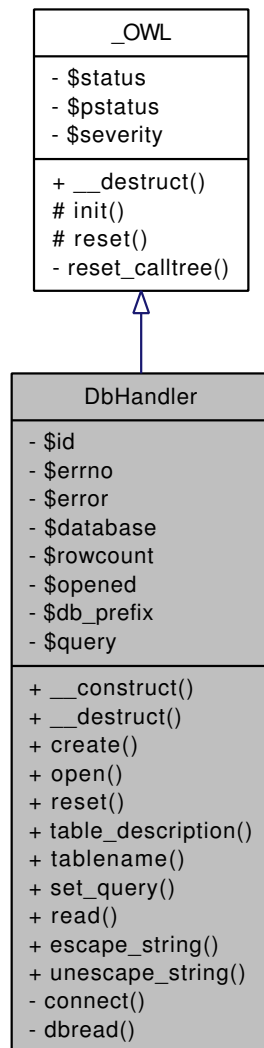
6.4 DbHandler Class Reference

Database handler.

Inheritance diagram for DbHandler:



Collaboration diagram for DbHandler:



Public Member Functions

- `__construct` (\$srv= 'localhost', \$db= "", \$usr= "", \$pwd= "", \$dbtype= 'MySQL')
- `__destruct` ()
- `create` ()
- `open` ()
- `reset` ()
- `table_description` (\$tablename, &\$data)
- `tablename` (\$tablename)
- `set_query` (\$qry)
- `read` (\$flag= `DBHANDLE_DATA`, &\$data, \$quick_query= "", \$line=0, \$file= '[unknown]')
- `escape_string` (\$string)
- `unescape_string` (\$string)

Protected Member Functions

- [init](#) ()

Private Member Functions

- [connect](#) ()
- [dbread](#) (\$qry, &\$rows, &\$fields)

Private Attributes

- [\\$id](#)
- [\\$errno](#)
- [\\$error](#)
- [\\$database](#)
- [\\$rowcount](#)
- [\\$opened](#)
- [\\$db_prefix](#)
- [\\$query](#)

6.4.1 Detailed Description

Database handler.

Handler for all database I/O. This class uses an (abstract) class for the actual storage.

Author:

Oscar van Eijk, Oveas Functionality Provider

Version:

May 15, 2007 – O van Eijk – initial version for Terra-Terra

Jul 29, 2008 – O van Eijk – Modified version for [OWL](#)

6.4.2 Constructor & Destructor Documentation

6.4.2.1 DbHandler::__construct (\$ *srv* = 'localhost', \$ *db* = '', \$ *usr* = '', \$ *pwd* = '', \$ *dbtype* = 'MySQL')

Class constructor; opens the database connection.

Parameters:

- ← *\$srv* Database server
- ← *\$db* Database name
- ← *\$usr* Username to connect with
- ← *\$pwd* Password to use for connection
- ← *\$dbtype* Database type (reserved for future use, currently only MySQL is implemented)

6.4.2.2 DbHandler::__destruct ()

Class destructor; closes the database connection

Reimplemented from [_OWL](#).

6.4.3 Member Function Documentation

6.4.3.1 DbHandler::create ()

Create a new database

Returns:

Severity level

6.4.3.2 DbHandler::connect () [private]

Connect to the database server

Returns:

True on success, otherwise False

6.4.3.3 DbHandler::open ()

Opens the database connection.

Returns:

Severity level

6.4.3.4 DbHandler::reset ()

Reset the object

Reimplemented from [_OWL](#).

6.4.3.5 DbHandler::table_description (\$ tablename, &\$ data)

Get a description of a database table

Parameters:

← *\$tablename* The tablename

→ *\$data* Indexed array holding all fields => datatypes

Returns:

Indexed array holding all fields => datatypes

6.4.3.6 DbHandler::tablename (\$ *tablename*)

Extend a tablename with the database prefix

Parameters:

← *\$tablename* Table name to extend

Returns:

Extended table name

6.4.3.7 DbHandler::set_query (\$ *qry*)

Set the database query. This function should only be used if the query is too complex to be set with any of the prepare() functions.

Parameters:

← *\$qry* A complete database query. All tablenames must be prefixed (with the [tablename\(\)](#) function)!

6.4.3.8 DbHandler::read (\$ *flag* = DBHANDLE_DATA, &\$ *data*, \$ *quick_query* = "", \$ *line* = 0, \$ *file* = '[unknown]')

Read from the database. The return value depends on the flag. By default, the selected rows(s) are returned in a 2d array.

Parameters:

← *\$flag* Flag that identifies how data should be returned; as data (default) or the number of rows

→ *\$data* The retrieved value in a formatt depending on the flag:

- DBHANDLE_ROWCOUNT; Number of matching rows
- DBHANDLE_FIELDCOUNT; Number of fields per rows
- DBHANDLE_TOTALFIELDcount; Total number op fields
- DBHANDLE_DATA (default); A 2D array with all data
- DBHANDLE_SINGLEROW; The first matching row in a 1D array
- DBHANDLE_SINGLEFIELD; The first matching field

- ← *\$quick_query* Database query string. If empty, `$this->query` is used
- ← *\$line* Line number of this call
- ← *\$file* File that made the call to this method

Returns:

Severity level

6.4.3.9 DbHandler::dbread (\$ qry, &\$ rows, &\$ fields) [private]

Read from the database.

Parameters:

- ← *\$qry* Database query string.
- *\$rows* Number of rows matched
- *\$fields* Number of fields per row

Returns:

A 2D array with all data, or false on failures

6.4.3.10 DbHandler::escape_string (\$ string)

Call the current DBtype's escape function for character strings.

Parameters:

\$string The string that should be escaped

Returns:

The escaped string

6.4.3.11 DbHandler::unescape_string (\$ string)

Call the current DBtype's unescape function for character strings.

Parameters:

\$string The string that should be unescaped

Returns:

The unescaped string

6.4.3.12 `__OWL::init ()` [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this **might** indicate something went wrong.

6.4.4 Member Data Documentation

6.4.4.1 `DbHandler::$id` [private]

integer - DB Handle ID

6.4.4.2 `DbHandler::$errno` [private]

integer - Error ID

6.4.4.3 `DbHandler::$error` [private]

string - Error text

6.4.4.4 `DbHandler::$database` [private]

array - Database location and authorization info

6.4.4.5 `DbHandler::$rowcount` [private]

integer - Row counter

6.4.4.6 `DbHandler::$opened` [private]

boolean - True if the database is opened

6.4.4.7 `DbHandler::$db_prefix` [private]

string - database prefix

6.4.4.8 `DbHandler::$query` [private]

string - Query string

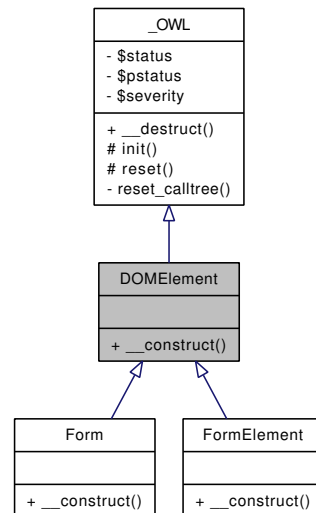
The documentation for this class was generated from the following file:

- `/home/oscar/work/eclipse/owl-php/src/kernel/so/class.dbhandler.php`

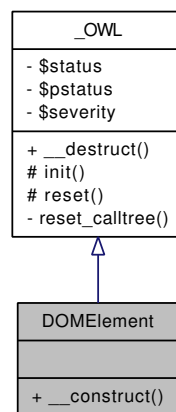
6.5 DOMElement Class Reference

DOM Element base class.

Inheritance diagram for DOMElement:



Collaboration diagram for DOMElement:



Public Member Functions

- `__construct()`

Protected Member Functions

- `init()`
- `reset()`

6.5.1 Detailed Description

DOM Element base class.

Base class for all DOM elements

Author:

Oscar van Eijk, Oveas Functionality Provider

Version:

Aug 29, 2008 – O van Eijk – initial version

6.5.2 Constructor & Destructor Documentation

6.5.2.1 DOMELEMENT::__construct ()

Class constructor;

Reimplemented in [Form](#), and [FormElement](#).

6.5.3 Member Function Documentation

6.5.3.1 __OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

6.5.3.2 __OWL::reset () [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#).

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

- [/home/oscar/work/eclipse/owl-php/src/kernel/ui/class.domelement.php](#)

6.6 FileHandler Class Reference

File handler.

Public Member Functions

- [__construct](#) (\$name= ”)
- [__destruct](#) ()
- [open](#) (\$mode= 'r')
- [close](#) ()
- [read_data](#) ()
- [read_line](#) (\$trim=FILE_NOTRIM)
- [encode](#) ()
- [download](#) ()

Public Attributes

- [\\$original_name](#)

6.6.1 Detailed Description

File handler.

Handle all files

Author:

Oscar van Eijk, Oveas Functionality Provider

Version:

May 15, 2007 – O van Eijk – initial version for Terra-Terra (based on an old OFM module)
Jul 30, 2008 – O van Eijk – Modified version for OWL-PHP

6.6.2 Constructor & Destructor Documentation

6.6.2.1 FileHandler::__construct (\$ *name* = ”)

6.6.2.2 FileHandler::__destruct ()

6.6.3 Member Function Documentation

6.6.3.1 `FileHandler::open ($ mode = 'r')`

6.6.3.2 `FileHandler::close ()`

6.6.3.3 `FileHandler::read_data ()`

6.6.3.4 `FileHandler::read_line ($ trim = FILE_NOTRIM)`

6.6.3.5 `FileHandler::encode ()`

6.6.3.6 `FileHandler::download ()`

6.6.4 Member Data Documentation

6.6.4.1 `FileHandler::$original_name`

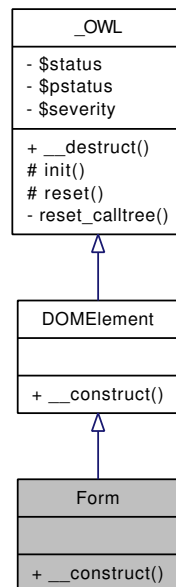
The documentation for this class was generated from the following file:

- `/home/oscar/work/eclipse/owl-php/src/kernel/so/class.filehandler.php`

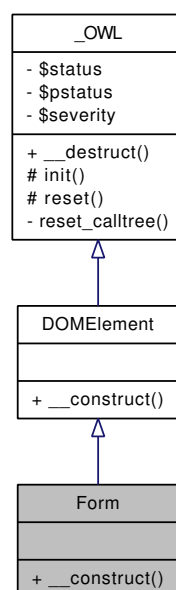
6.7 Form Class Reference

[Form](#) Element class.

Inheritance diagram for Form:



Collaboration diagram for Form:



Public Member Functions

- [__construct](#) ()

Protected Member Functions

- [init](#) ()
- [reset](#) ()

6.7.1 Detailed Description

[Form](#) Element class.

Define an HTML [Form](#). This class extends the [DOMElement](#) base class and implements the [FormElement](#) class

Author:

Oscar van Eijk, Oveas Functionality Provider

Version:

Aug 29, 2008 – O van Eijk – initial version

6.7.2 Constructor & Destructor Documentation

6.7.2.1 Form::__construct ()

Class constructor;

Reimplemented from [DOMElement](#).

6.7.3 Member Function Documentation

6.7.3.1 __OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this **might** indicate something went wrong.

6.7.3.2 __OWL::reset () [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#).

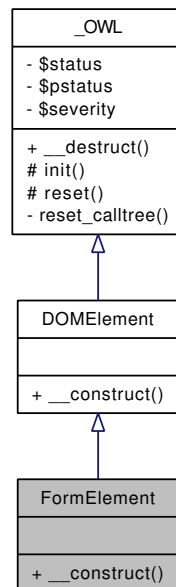
The documentation for this class was generated from the following file:

- [/home/oscar/work/eclipse/owl-php/src/kernel/bo/class.form.php](#)

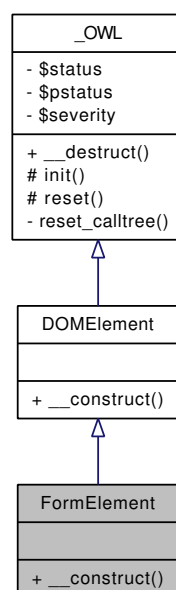
6.8 FormElement Class Reference

Form element.

Inheritance diagram for FormElement:



Collaboration diagram for FormElement:



Public Member Functions

- [__construct](#) ()

Protected Member Functions

- [init](#) ()
- [reset](#) ()

6.8.1 Detailed Description

[Form](#) element.

[Form](#) element

Author:

Oscar van Eijk, Oveas Functionality Provider

Version:

Aug 29, 2008 – O van Eijk – initial version

6.8.2 Constructor & Destructor Documentation

6.8.2.1 FormElement::__construct ()

Class constructor;

Reimplemented from [DOMElement](#).

6.8.3 Member Function Documentation

6.8.3.1 __OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this **might** indicate something went wrong.

6.8.3.2 __OWL::reset () [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#).

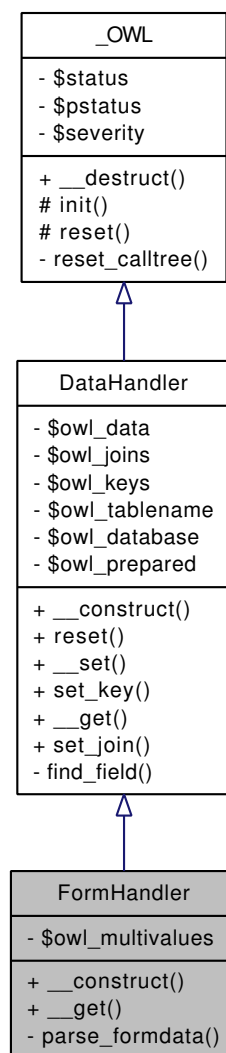
The documentation for this class was generated from the following file:

- [/home/oscar/work/eclipse/owl-php/src/kernel/ui/class.formelement.php](#)

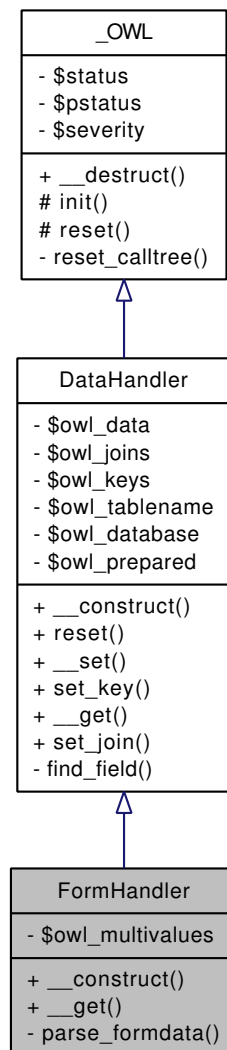
6.9 FormHandler Class Reference

Formhandler.

Inheritance diagram for FormHandler:



Collaboration diagram for FormHandler:



Public Member Functions

- [__construct](#) ()
- [__get](#) (\$variable)
- [reset](#) (\$level=[DATA_RESET_PREPARE](#))
- [__set](#) (\$variable, \$value)
- [set_key](#) (\$variable)
- [set_join](#) (\$lvalue, \$rvalue, \$linktype= '=')

Protected Member Functions

- [reset](#) ()
- [init](#) ()

Private Member Functions

- `parse_formdata` (\$data=null)

Private Attributes

- `$owl_multivalues`

6.9.1 Detailed Description

Formhandler.

Handler for all incoming formdata. It implements the datahandler class.

Author:

Oscar van Eijk, Oveas Functionality Provider

Version:

Aug 28, 2008 – O van Eijk – initial version

6.9.2 Constructor & Destructor Documentation

6.9.2.1 FormHandler::__construct ()

Class constructor; Parse the incoming formdata.

6.9.3 Member Function Documentation

6.9.3.1 FormHandler::parse_formdata (\$ *data* = null) [private]

Parse a given form ans store all data in the parent class, except values that come from a multiple select; they will be stored locally.

Parameters:

← *\$data* The formdata array.

6.9.3.2 FormHandler::__get (\$ *variable*)

Reimplement of the __get magic method; the parent's __get will only be called of the requested variable name is not in the 'local' array where multi-values are stored.

Parameters:

← *\$variable* The variable name who's value should be returned

Returns:

Value as taken form the form.

Reimplemented from [DataHandler](#).

6.9.3.3 DataHandler::reset (\$ level = DATA_RESET_PREPARE) [inherited]

Reset the object

Parameters:

← *\$level* The reset level, can be any of the following:

- DATA_RESET_STATUS; Reset object status only
- DATA_RESET_PREPARE (default); Reset status and a prepared query
- DATA_RESET_META; Reset status and query, remove all locks and joins
- DATA_RESET_FULL; Reset status and query, remove all locks, joins and data

6.9.3.4 __OWL::reset () [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#).

6.9.3.5 DataHandler::__set (\$ variable, \$ value) [inherited]

Define or override a variable in the data array

Parameters:

← *\$variable* The name of the variable that should be set

← *\$value* Value to set the variable to. If this is an array, the second field in the array has to be the tablename where the fieldname is found. The value itself van NEVER be an array!

6.9.3.6 DataHandler::set_key (\$ variable) [inherited]

Lock variables for update by adding them to an array. Fields in this array will not be overwritten on updates, but used in WHERE clauses.

Parameters:

← *\$variable* Variable name to lock, optionally as an array (table, field)

Returns:

Severity level

6.9.3.7 DataHandler::set_join (\$ lvalue, \$ rvalue, \$ linktype = '=') [inherited]

Define a link between 2 fields that will be recognized when the database query is built.

Parameters:

- ← *\$lvalue* Left value as array(table, field)
- ← *\$rvalue* Right value as array(table, field)
- ← *\$linktype* How are the fields linked. Can be any binary operator as recognized by SQL.

Returns:

Severity level

6.9.3.8 __OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this **might** indicate something went wrong.

6.9.4 Member Data Documentation

6.9.4.1 FormHandler::\$owl_multivalues [private]

Create an array which will hold multi-select values

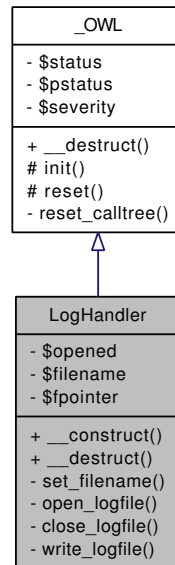
The documentation for this class was generated from the following file:

- /home/oscar/work/eclipse/owl-php/src/kernel/so/[class.formhandler.php](#)

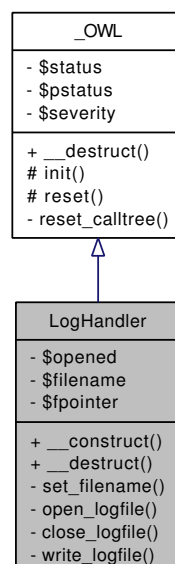
6.10 LogHandler Class Reference

Log handler.

Inheritance diagram for LogHandler:



Collaboration diagram for LogHandler:



Public Member Functions

- [__construct](#) ()
- [__destruct](#) ()

Protected Member Functions

- [init](#) ()
- [reset](#) ()

Private Member Functions

- [set_filename](#) ()
- [open_logfile](#) ()
- [close_logfile](#) ()
- [write_logfile](#) (\$msg)

Private Attributes

- [\\$opened](#)
- [\\$filename](#)
- [\\$pointer](#)

6.10.1 Detailed Description

Log handler.

This class handles all [OWL](#) logging

Author:

Oscar van Eijk, Oveas Functionality Provider

Version:

Aug 13, 2008 – O van Eijk – initial version

6.10.2 Constructor & Destructor Documentation

6.10.2.1 LogHandler::__construct ()

Class constructor

6.10.2.2 LogHandler::__destruct ()

Class destructor

Reimplemented from [_OWL](#).

6.10.3 Member Function Documentation

6.10.3.1 `LogHandler::set_filename ()` [private]

Find out what the filename of the logfile should be

6.10.3.2 `LogHandler::open_logfile ()` [private]

Open the logfile for write

6.10.3.3 `LogHandler::close_logfile ()` [private]

Close the logfile

6.10.3.4 `LogHandler::write_logfile ($ msg)` [private]

Write a message to the logfile

Parameters:

← *\$msg* The complete log message

6.10.3.5 `_OWL::init ()` [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this **might** indicate something went wrong.

6.10.3.6 `_OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#).

6.10.4 Member Data Documentation

6.10.4.1 `LogHandler::$opened` [private]

Boolean to keep track of the logfile status

6.10.4.2 `LogHandler::$filename` [private]

Name of the logfile

6.10.4.3 LogHandler::\$fpointer [private]

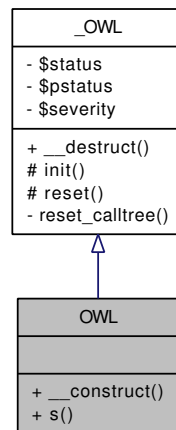
File pointer

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

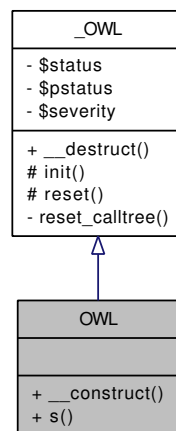
- [/home/oscar/work/eclipse/owl-php/src/kernel/so/class.loghandler.php](#)

6.11 OWL Class Reference

Inheritance diagram for OWL:



Collaboration diagram for OWL:



Public Member Functions

- `__construct` ()
- `s` (\$a, \$b=array())

Protected Member Functions

- `init` ()
- `reset` ()

6.11.1 Constructor & Destructor Documentation

6.11.1.1 OWL::__construct ()

Constructor

6.11.2 Member Function Documentation

6.11.2.1 OWL::s (\$ a, \$ b = array())

Call to set_status()

Parameters:

- ← *\$a* First parameter for passthrough
- ← *\$b* Second parameter for passthrough

6.11.2.2 __OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this **might** indicate something went wrong.

6.11.2.3 __OWL::reset () [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#).

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

- [/home/oscar/work/eclipse/owl-php/src/kernel/class._OWL.php](#)

6.12 OWLException Class Reference

Exception handler.

Public Member Functions

- [__construct](#) (\$msg=null, \$code=0, [OWLException](#) \$caller=null)
- [get_trace](#) ()
- [stack_dump](#) (\$textmode=true)

Public Attributes

- [\\$thrown_code](#)

Private Member Functions

- [traceback](#) (\$trace, \$step, \$textmode)

Private Attributes

- [\\$caller](#)

6.12.1 Detailed Description

Exception handler.

Extend the default PHP Exception handler .

Author:

Oscar van Eijk, Oveas Functionality Provider

Version:

Jul 29, 2008 – O van Eijk – Initial version

6.12.2 Constructor & Destructor Documentation

6.12.2.1 OWLException::__construct (\$ *msg* = null, \$ *code* = 0, OWLException \$ *caller* = null)

Create the Exception handler object

Parameters:

← *\$msg* Message text

- ← *\$code* Code of the event
- ← *\$caller* Backlink to the calling object

6.12.3 Member Function Documentation

6.12.3.1 OWLException::get_trace ()

Trace back the previous object in the stack

6.12.3.2 OWLException::stack_dump (\$ *textmode* = true)

Create an overview of the calling stack

Parameters:

- ← *\$textmode* specify the format in which the stackdump should be returned; text (true, default) or HTML (false).

6.12.3.3 OWLException::traceback (\$ *trace*, \$ *step*, \$ *textmode*) [private]

Trace a single call from the stack

Parameters:

- ← *\$trace* The call
- ← *\$step* Number of the call
- ← *\$textmode* True if the stackdump is created as ASCII text, False for HTML

6.12.4 Member Data Documentation

6.12.4.1 OWLException::\$caller [private]

Backlink to the calling object

6.12.4.2 OWLException::\$thrown_code

Store the error code to allow the logger to retrieve it

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

- /home/oscar/work/eclipse/owl-php/src/kernel/so/[class.exceptionhandler.php](#)

6.13 OWLExceptionHandler Class Reference

Default exception handler.

Static Public Member Functions

- static [log_exception](#) ([OWLException](#) \$exception)
- static [handle_exception](#) ([OWLException](#) \$exception)

6.13.1 Detailed Description

Default exception handler.

Establish a default exception handler. En exception is thrown whenever a status is set above a specified severity level. These exceptions are caught only by this default handler

Author:

Oscar van Eijk, Oveas Functionality Provider

Version:

Jul 29, 2008 – O van Eijk – Initial version

6.13.2 Member Function Documentation

6.13.2.1 static OWLExceptionHandler::log_exception (OWLException \$ *exception*) [static]

Show the stackdump of an exception

Parameters:

← *\$exception* The exception

6.13.2.2 static OWLExceptionHandler::handle_exception (OWLException \$ *exception*) [static]

Catch an uncaught exception

Parameters:

← *\$exception* The exception

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

- /home/oscar/work/eclipse/owl-php/src/kernel/so/[class.exceptionhandler.php](#)

6.14 Register Class Reference

Static Public Member Functions

- static `init` ()
- static `register_app` (\$name, \$id)
- static `register_class` (\$name)
- static `register_code` (\$code)
- static `register_severity` (\$level, \$name)
- static `get_severity` (\$level)
- static `get_severity_level` (\$name)
- static `get_run_id` ()
- static `get_code` (\$value)
- static `set_application` (\$app_id)
- static `set_class` (\$class_id)
- static `set_severity` (\$severity_level)

6.14.1 Detailed Description

OWL keeps track of all running applications, their class and all status codes their instances (objects) can have. This is done in a global [Register](#), which is maintained by this class.

6.14.2 Member Function Documentation

6.14.2.1 static `Register::init` () [static]

Initialise the register array

6.14.2.2 static `Register::register_app` (\$ *name*, \$ *id*) [static]

Store the specified application in the register

Parameters:

- ← *\$name* Name of the class
- ← *\$id* Application ID

6.14.2.3 static Register::register_class (\$ name) [static]

Store the specified class in the register, and setup an array to keep track of the codes

Parameters:

← *\$name* Name of the class

6.14.2.4 static Register::register_code (\$ code) [static]

Define a new statuscode

Parameters:

← *\$code* Symbolic name of the status code

6.14.2.5 static Register::register_severity (\$ level, \$ name) [static]

Store the known severitylevels in the register

Parameters:

← *\$level* Symbolic name for the severity level

← *\$name* Human readable value

6.14.2.6 static Register::get_severity (\$ level) [static]

Read a severity level from the register

Parameters:

← *\$level* Hex value of the severity level

Returns:

Human readable value

6.14.2.7 static Register::get_severity_level (\$ name) [static]

This function is used by a config parse to translate a string value to the appropriate severity level

Parameters:

← *\$name* The name of the severity level

Returns:

Hex value of the severity level

6.14.2.8 static Register::get_run_id () [static]

Return the ID of the current run

6.14.2.9 static Register::get_code (\$ value) [static]

Translate an hex value code to the symbolic name

Parameters:

← *\$value* Hex value of the status code

Returns:

Human readable value

6.14.2.10 static Register::set_application (\$ app_id) [static]

Point the register to the specified application.

Parameters:

← *\$app_id* Application ID

6.14.2.11 static Register::set_class (\$ class_id) [static]

Point the register to the specified class.

Parameters:

← *\$class_id* Class ID

6.14.2.12 static Register::set_severity (\$ severity_level) [static]

Set the current severity to the specified level in the [Register](#)

Parameters:

← *\$severity_level* Severity level

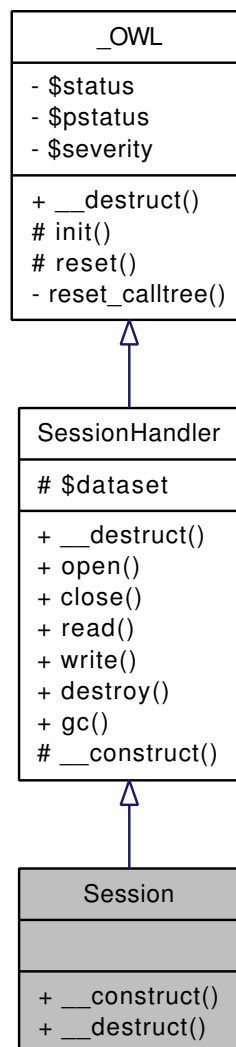
The documentation for this class was generated from the following file:

- /home/oscar/work/eclipse/owl-php/src/kernel/so/[class.register.php](#)

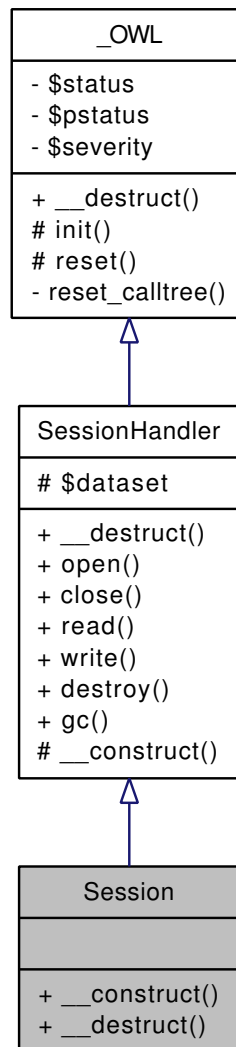
6.15 Session Class Reference

the OWL-PHP session object

Inheritance diagram for Session:



Collaboration diagram for Session:



Public Member Functions

- `__construct ()`
- `__destruct ()`
- `open ()`
- `close ()`
- `read ($id)`
- `write ($id, $data)`
- `destroy ($id)`
- `gc ($lifetime)`

Protected Member Functions

- `init ()`
- `reset ()`

Protected Attributes

- [\\$dataset](#)

6.15.1 Detailed Description

the OWL-PHP session object

This class handles the [OWL](#) session

Author:

Oscar van Eijk, Oveas Functionality Provider

Version:

Aug 13, 2008 – O van Eijk – initial version

6.15.2 Constructor & Destructor Documentation

6.15.2.1 Session::__construct ()

When a new run is initialised, restore an older session or create a new one

Reimplemented from [SessionHandler](#).

6.15.2.2 Session::__destruct ()

When a run ends, write the sessiondata

Reimplemented from [SessionHandler](#).

6.15.3 Member Function Documentation

6.15.3.1 SessionHandler::open () [inherited]

Open the session

Returns:

bool

6.15.3.2 SessionHandler::close () [inherited]

Close the session

Returns:

bool

6.15.3.3 SessionHandler::read (\$ *id*) [inherited]

Read the session

Parameters:

← *\$id* session id

Returns:

[Session](#) data

6.15.3.4 SessionHandler::write (\$ *id*, \$ *data*) [inherited]

Write the session

Parameters:

← *\$id* [Session](#) id

← *\$data* [Session](#) data

Returns:

True on success, Failure otherwise

6.15.3.5 SessionHandler::destroy (\$ *id*) [inherited]

Destroy a session

Parameters:

← *\$id* ID of the session to destroy

Returns:

Boolean indicating success (true) or failure (false)

6.15.3.6 SessionHandler::gc (\$ *lifetime*) [inherited]

Garbage Collector. By default, this has a 1% change of being executed (session.gc_probability/session.gc_divisor). The values can be changed in the constructor.

Parameters:

← *\$lifetime* [Session](#) lifetime in seconds. By default, 1440 seconds. This can be changes in the constructor

Returns:

Status code

6.15.3.7 `__OWL::init ()` [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this **might** indicate something went wrong.

6.15.3.8 `__OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#).

6.15.4 Member Data Documentation**6.15.4.1** `SessionHandler::$dataset` [protected, inherited]

Link to a datahandler object. This dataset is used as an interface to all database IO.

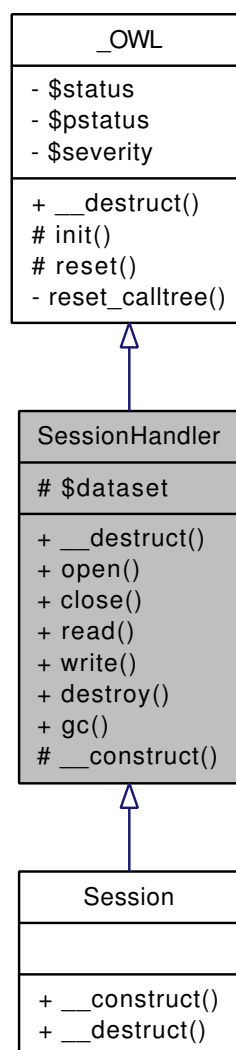
The documentation for this class was generated from the following file:

- `/home/oscar/work/eclipse/owl-php/src/kernel/bo/class.session.php`

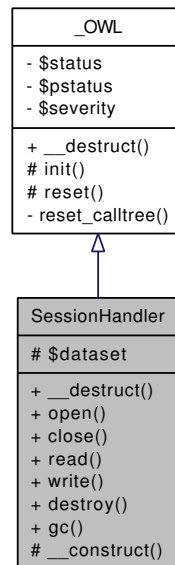
6.16 SessionHandler Class Reference

the PHP session object

Inheritance diagram for SessionHandler:



Collaboration diagram for SessionHandler:



Public Member Functions

- [__destruct](#) ()
- [open](#) ()
- [close](#) ()
- [read](#) (\$id)
- [write](#) (\$id, \$data)
- [destroy](#) (\$id)
- [gc](#) (\$lifetime)

Protected Member Functions

- [__construct](#) ()
- [init](#) ()
- [reset](#) ()

Protected Attributes

- [\\$dataset](#)

6.16.1 Detailed Description

the PHP session object

This class saves and (re)stores the user sessions

Author:

Oscar van Eijk, Oveas Functionality Provider

Version:

Jul 29, 2008 – O van Eijk – initial version

6.16.2 Constructor & Destructor Documentation

6.16.2.1 SessionHandler::__construct () [protected]

Class constructor; set the save_handler

Reimplemented in [Session](#).

6.16.2.2 SessionHandler::__destruct ()

Write the session data back to the database

Reimplemented from [_OWL](#).

Reimplemented in [Session](#).

6.16.3 Member Function Documentation

6.16.3.1 SessionHandler::open ()

Open the session

Returns:

bool

6.16.3.2 SessionHandler::close ()

Close the session

Returns:

bool

6.16.3.3 SessionHandler::read (\$ id)

Read the session

Parameters:

← *\$id* session id

Returns:

[Session](#) data

6.16.3.4 SessionHandler::write (\$ *id*, \$ *data*)

Write the session

Parameters:

- ← *\$id* [Session](#) id
- ← *\$data* [Session](#) data

Returns:

True on success, Failure otherwise

6.16.3.5 SessionHandler::destroy (\$ *id*)

Destroy a session

Parameters:

- ← *\$id* ID of the session to destroy

Returns:

Boolean indicating success (true) or failure (false)

6.16.3.6 SessionHandler::gc (\$ *lifetime*)

Garbage Collector. By default, this has a 1% change of being executed (session.gc_probability/session.gc_divisor). The values can be changed in the constructor.

Parameters:

- ← *\$lifetime* [Session](#) lifetime in seconds. By default, 1440 seconds. This can be changes in the constructor

Returns:

Status code

6.16.3.7 __OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this **might** indicate something went wrong.

6.16.3.8 `__OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#).

6.16.4 Member Data Documentation

6.16.4.1 `SessionHandler::$dataset` [protected]

Link to a datahandler object. This dataset is used as an interface to all database IO.

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

- </home/oscar/work/eclipse/owl-php/src/kernel/so/class.sessionhandler.php>

6.17 StatusHandler Class Reference

Status object.

Public Member Functions

- [__construct](#) (\$code=OWL_STATUS_WARNING)
- [set_code](#) (\$code=OWL_STATUS_BUG)
- [reset](#) ()
- [set_params](#) (\$params)
- [get_severity](#) (\$status=null)
- [get_code](#) ()
- [get_message](#) ()

Private Attributes

- [\\$code](#)
- [\\$params](#)

6.17.1 Detailed Description

Status object.

Each object, when initialised, gets a status object which olds information about the last action that was performed.

Author:

Oscar van Eijk, Oveas Functionality Provider

Version:

Aug 11, 2008 – O van Eijk – Initial version

6.17.2 Constructor & Destructor Documentation

6.17.2.1 StatusHandler::__construct (\$ code = OWL_STATUS_WARNING)

Constructor; should be called only by [_OWL::init\(\)](#). The default status is initially a (generic) warning status. It should be set to any successfull status after object initialisation completed.

Parameters:

← *\$code* The status code

6.17.3 Member Function Documentation

6.17.3.1 StatusHandler::set_code (\$ *code* = OWL_STATUS_BUG)

Set the status of the owner object to the given value.

Parameters:

← *\$code* The status code

Returns:

The severity level of the code

6.17.3.2 StatusHandler::reset ()

Reset the object status

6.17.3.3 StatusHandler::set_params (\$ *params*)

If the status was set with optional parameters, they will be set in this subject and substituted in the correct message

Parameters:

← *\$params* An array with parameters

6.17.3.4 StatusHandler::get_severity (\$ *status* = null)

Check the status of the given object and return its severity.

Parameters:

← *\$status* An optional parameter to check an other status code i.s.o the object's current status.

Returns:

The severity level of the current status

6.17.3.5 StatusHandler::get_code ()

Return the status code

Returns:

The status code

6.17.3.6 StatusHandler::get_message ()

Retrieve the message that has been specified for the current object status. If parameters are set and wanted by the message text, replace them. If more parameters are available then used in the message, there's a quote with the unused parameter count added to the message.

Returns:

The message text

6.17.4 Member Data Documentation**6.17.4.1 StatusHandler::\$code [private]**

Current object status

6.17.4.2 StatusHandler::\$params [private]

Array with parameters that will be substituted in message text

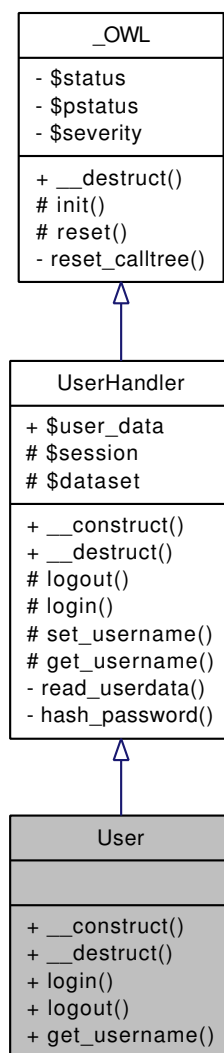
The documentation for this class was generated from the following file:

- </home/oscar/work/eclipse/owl-php/src/kernel/so/class.statushandler.php>

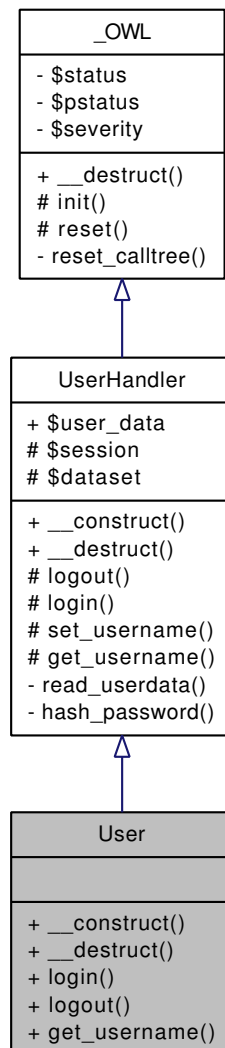
6.18 User Class Reference

the OWL-PHP session object

Inheritance diagram for User:



Collaboration diagram for User:



Public Member Functions

- `__construct` (\$username=null)
- `__destruct` ()
- `login` (\$username, \$password)
- `logout` ()
- `get_username` ()

Public Attributes

- `$user_data`

Protected Member Functions

- `login` (\$password)

- [set_username](#) (\$username)
- [init](#) ()
- [reset](#) ()

Protected Attributes

- [\\$session](#)
- [\\$dataset](#)

6.18.1 Detailed Description

the OWL-PHP session object

This class handles the [OWL](#) users

Author:

Oscar van Eijk, Oveas Functionality Provider

Version:

Aug 27, 2008 – O van Eijk – initial version

6.18.2 Constructor & Destructor Documentation

6.18.2.1 User::__construct (\$ *username* = null)

Class constructor; create a new user environment

Parameters:

← *\$username* Username when logged in. Default username is 'anonymous'

Reimplemented from [UserHandler](#).

6.18.2.2 User::__destruct ()

When a run ends, write the sessiondata

Reimplemented from [UserHandler](#).

6.18.3 Member Function Documentation

6.18.3.1 User::login (\$ *username*, \$ *password*)

Log in

Parameters:

- ← *\$username* Given username
- ← *\$password* Given password

Returns:

True on success, False otherwise

6.18.3.2 User::logout ()

Log out the current user

Reimplemented from [UserHandler](#).

6.18.3.3 User::get_username ()

Return the username of the current session

Reimplemented from [UserHandler](#).

6.18.3.4 UserHandler::login (\$ *password*) [protected, inherited]

Attempt to log in with the current user and the given password

Parameters:

- ← *\$password* The user provided password

Returns:

True on success, False otherwise

6.18.3.5 UserHandler::set_username (\$ *username*) [protected, inherited]

Set the username

Parameters:

- ← *\$username* Username

6.18.3.6 __OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this **might** indicate something went wrong.

6.18.3.7 `_OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#).

6.18.4 Member Data Documentation

6.18.4.1 `UserHandler::$session` [protected, inherited]

The PHP session object

6.18.4.2 `UserHandler::$dataset` [protected, inherited]

Link to a datahandler object. This dataset is used as an interface to all database IO.

6.18.4.3 `UserHandler::$user_data` [inherited]

An indexed array with the user information as take from the database

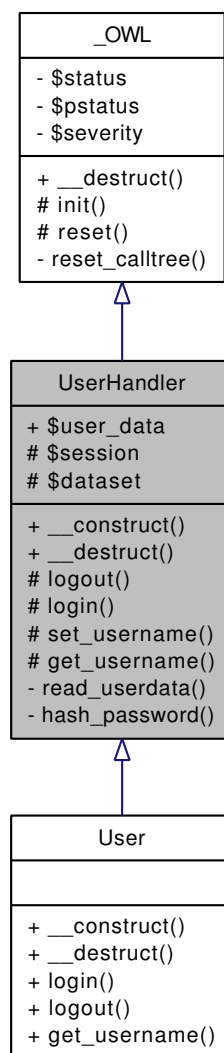
The documentation for this class was generated from the following file:

- [/home/oscar/work/eclipse/owl-php/src/kernel/bo/class.user.php](#)

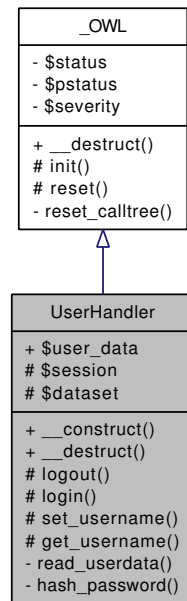
6.19 UserHandler Class Reference

the user object

Inheritance diagram for UserHandler:



Collaboration diagram for UserHandler:



Public Member Functions

- `__construct` (\$username)
- `__destruct` ()

Public Attributes

- `$user_data`

Protected Member Functions

- `logout` ()
- `login` (\$password)
- `set_username` (\$username)
- `get_username` ()
- `init` ()
- `reset` ()

Protected Attributes

- `$session`
- `$dataset`

Private Member Functions

- `read_userdata` ()
- `hash_password` (\$password)

6.19.1 Detailed Description

the user object

This class creates user sessions and handles logging in and out

Author:

Oscar van Eijk, Oveas Functionality Provider

Version:

Aug 27, 2008 – O van Eijk – initial version

6.19.2 Constructor & Destructor Documentation

6.19.2.1 `UserHandler::__construct ($ username)`

Class constructor; setup the user environment

Parameters:

← *\$username* Username.

Reimplemented in [User](#).

6.19.2.2 `UserHandler::__destruct ()`

Class destructor

Reimplemented from [_OWL](#).

Reimplemented in [User](#).

6.19.3 Member Function Documentation

6.19.3.1 `UserHandler::logout ()` [protected]

Log out the current user

Reimplemented in [User](#).

6.19.3.2 `UserHandler::login ($ password)` [protected]

Attempt to log in with the current user and the given password

Parameters:

← *\$password* The user provided password

Returns:

True on success, False otherwise

6.19.3.3 UserHandler::read_userdata () [private]

When a new session starts for a use that was logged in before retrieve the userdata back from the database

6.19.3.4 UserHandler::hash_password (\$ password) [private]

Encrypt a given password

Parameters:

← *\$password* Given password in plain text format

Returns:

The encrypted password

6.19.3.5 UserHandler::set_username (\$ username) [protected]

Set the username

Parameters:

← *\$username* Username

6.19.3.6 UserHandler::get_username () [protected]

Return the username of the current session

Reimplemented in [User](#).

6.19.3.7 _OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

6.19.3.8 _OWL::reset () [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#).

6.19.4 Member Data Documentation

6.19.4.1 `UserHandler::$session` [protected]

The PHP session object

6.19.4.2 `UserHandler::$dataset` [protected]

Link to a datahandler object. This dataset is used as an interface to all database IO.

6.19.4.3 `UserHandler::$user_data`

An indexed array with the user information as take from the database

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

- `/home/oscar/work/eclipse/owl-php/src/kernel/so/class.userhandler.php`

Chapter 7

OWL-PHP File Documentation

7.1 /home/oscar/work/eclipse/owl-php/src/config.php File Reference

Variables

- `$GLOBALS` ['register']
- `$GLOBALS` ['config']['configfiles']['owl'] = `OWL_ROOT` . '/owl_config.cfg'
- `$GLOBALS` ['config']['hide']['tag'] = '(hide)'
- `$GLOBALS` ['config']['hide']['value'] = '(hidden)'

7.1.1 Detailed Description

Configuration basics file for `OWL`; it stores some fixed configuration in a global data structure

Version:

Id

`config.php`, v 1.3 2008-08-28 18:12:52 oscar Exp

7.1.2 Variable Documentation

7.1.2.1 `$GLOBALS['config']['hide']['value'] = '(hidden)'`

7.1.2.2 `$GLOBALS['config']['hide']['tag'] = '(hide)'`

7.1.2.3 `$GLOBALS['config']['configfiles']['owl'] = OWL_ROOT . '/owl_config.cfg'`

7.1.2.4 `$GLOBALS['register']`

Initial value:

```
array (
    'applications' => array()
    , 'classes' => array()
)
```

7.2 /home/oscar/work/eclipse/owl-php/src/index.php File Reference

Enumerations

- enum [OWL_ROOT](#)

7.2.1 Detailed Description

This is the entry point for OWL-PHP teststub

Version:

Id

[index.php](#), v 1.4 2008-09-02 05:16:53 oscar Exp

7.2.2 Enumeration Type Documentation

7.2.2.1 enum OWL_ROOT

7.3 /home/oscar/work/eclipse/owl-php/src/kernel/bo/class.form.php File Reference

Classes

- class [Form](#)
[Form](#) Element class.

7.3.1 Detailed Description

This file defines the HTML [Form](#) class

Version:

Id

[class.form.php](#),v 1.1 2008-09-02 05:16:54 oscar Exp

7.4 /home/oscar/work/eclipse/owl-php/src/kernel/bo/class.session.php File Reference

Classes

- class [Session](#)
the OWL-PHP session object

7.4.1 Detailed Description

This file defines the [Session](#) class

Version:

Id

[class.session.php](#),v 1.2 2008-08-28 18:12:52 oscar Exp

7.5 /home/oscar/work/eclipse/owl-php/src/kernel/bo/class.user.php File Reference

Classes

- class [User](#)
the OWL-PHP session object

7.5.1 Detailed Description

This file defines the [User](#) class

Version:

Id

[class.user.php](#),v 1.1 2008-08-28 18:12:52 oscar Exp

7.6 /home/oscar/work/eclipse/owl-php/src/kernel/class._ - OWL.php File Reference

Classes

- class [_OWL](#)
- class [OWL](#)

7.6.1 Detailed Description

This file defines the Oveas Web Library main class

Version:

Id

[class._OWL.php](#),v 1.3 2008-09-02 05:16:54 oscar Exp

7.7 `/home/oscar/work/eclipse/owl-php/src/kernel/so/class.confighandler` File Reference

Classes

- class [ConfigHandler](#)
Configuration handler.

7.7.1 Detailed Description

Define a class for config handling

Version:

Id

[class.confighandler.php](#), v 1.2 2008-08-28 18:12:52 oscar Exp

7.8 /home/oscar/work/eclipse/owl-php/src/kernel/so/class.datahandler.php File Reference

Classes

- class [DataHandler](#)

*The **OWL** Data object.*

Enumerations

Query preparation tools

These flags define what type of queries can be prepared

- enum [DATA_UNPREPARED](#)
Default value; no query prepared yet.
- enum [DATA_READ](#)
Read data from the database.
- enum [DATA_WRITE](#)
Write new data to the database.
- enum [DATA_UPDATE](#)
Update data in the database.
- enum [DATA_DELETE](#)
Remove data from the database.

Reset flags

These flags how an object should be performed. All values includes all lower values as well!

- enum [DATA_RESET_STATUS](#)
Reset object status only.
- enum [DATA_RESET_PREPARE](#)
Reset prepared queries as well.
- enum [DATA_RESET_META](#)
Remove all locks and joins.
- enum [DATA_RESET_FULL](#)
Clean all data.

7.8.1 Detailed Description

This file defines the [DataHandler](#) class

Version:

Id

[class.datahandler.php](#),v 1.2 2008-08-28 18:12:52 oscar Exp

7.8.2 Enumeration Type Documentation

7.8.2.1 enum DATA_DELETE

Remove data from the database.

7.8.2.2 enum DATA_READ

Read data from the database.

7.8.2.3 enum DATA_RESET_FULL

Clean all data.

7.8.2.4 enum DATA_RESET_META

Remove all locks and joins.

7.8.2.5 enum DATA_RESET_PREPARE

Reset prepared queries as well.

7.8.2.6 enum DATA_RESET_STATUS

Reset object status only.

7.8.2.7 enum DATA_UNPREPARED

Default value; no query prepared yet.

7.8.2.8 enum DATA_UPDATE

Update data in the database.

7.8.2.9 enum DATA_WRITE

Write new data to the database.

7.9 /home/oscar/work/eclipse/owl-php/src/kernel/so/class.dbhandler.p File Reference

Classes

- class [DbHandler](#)
Database handler.

Enumerations

Return Flags

These flags that define what value should be returned by read()

- enum [DBHANDLE_DATA](#)
Return the read values (default).
- enum [DBHANDLE_SINGLEFIELD](#)
Return the read value as a single field (i.s.o. a 2D array).
- enum [DBHANDLE_SINGLEROW](#)
Return the read value as a single row (a 1D array).
- enum [DBHANDLE_ROWCOUNT](#)
Return the number of rows.
- enum [DBHANDLE_FIELDCOUNT](#)
Return the number of fields per row.
- enum [DBHANDLE_TOTALFIELDCOUNT](#)
Return the total number of fields.

7.9.1 Detailed Description

This file defines the Database Handler class

Version:

Id

[class.dbhandler.php](#),v 1.2 2008-08-28 18:12:52 oscar Exp

7.9.2 Enumeration Type Documentation

7.9.2.1 enum DBHANDLE_DATA

Return the read values (default).

7.9.2.2 enum DBHANDLE_FIELDCOUNT

Return the number of fields per row.

7.9.2.3 enum DBHANDLE_ROWCOUNT

Return the number of rows.

7.9.2.4 enum DBHANDLE_SINGLEFIELD

Return the read value as a single field (i.s.o. a 2D array).

7.9.2.5 enum DBHANDLE_SINGLEROW

Return the read value as a single row (a 1D array).

7.9.2.6 enum DBHANDLE_TOTALFIELDCOUNT

Return the total number of fields.

7.10 `/home/oscar/work/eclipse/owl-php/src/kernel/so/class.exceptionh` File Reference

Classes

- class [OWLException](#)
Exception handler.
- class [OWLExceptionHandler](#)
Default exception handler.

7.10.1 Detailed Description

This file defines the [OWL](#) Exception handler class and a default exception handler, for which a special class is created.

Version:

Id

[class.exceptionhandler.php](#),v 1.3 2008-09-02 05:16:53 oscar Exp

7.11 /home/oscar/work/eclipse/owl-php/src/kernel/so/class.filehandler.php File Reference

Classes

- class [FileHandler](#)
File handler.

Enumerations

- enum [FILE_NOTRIM](#)
- enum [FILE_TRIM_L](#)
- enum [FILE_TRIM_R](#)
- enum [FILE_TRIM_C](#)

7.11.1 Detailed Description

This file defines the [FileHandler](#) class

Version:

Id

[class.filehandler.php](#), v 1.2 2008-08-28 18:12:52 oscar Exp

7.11.2 Enumeration Type Documentation

7.11.2.1 enum FILE_NOTRIM

7.11.2.2 enum FILE_TRIM_C

7.11.2.3 enum FILE_TRIM_L

7.11.2.4 enum FILE_TRIM_R

7.12 `/home/oscar/work/eclipse/owl-php/src/kernel/so/class.formhandl` File Reference

Classes

- class [FormHandler](#)
Formhandler.

7.12.1 Detailed Description

This file defines the Formhandler class

Version:

Id

[class.formhandler.php](#), v 1.2 2008-09-02 05:16:53 oscar Exp

7.13 `/home/oscar/work/eclipse/owl-php/src/kernel/so/class.loghandler.php` File Reference

Classes

- class [LogHandler](#)
Log handler.

7.13.1 Detailed Description

This file defines the Loghandler class

Version:

Id

[class.loghandler.php](#), v 1.3 2008-09-02 05:16:53 oscar Exp

7.14 /home/oscar/work/eclipse/owl-php/src/kernel/so/class.register.php File Reference

Classes

- class [Register](#)

Enumerations

Bitmaps

These bitmaps define the layout of status codes. They are used to extract information from the code

- enum [OWL_APPLICATION_PATTERN](#)
- enum [OWL_OBJECT_PATTERN](#)
- enum [OWL_STATUS_PATTERN](#)
- enum [OWL_SEVERITY_PATTERN](#)

7.14.1 Detailed Description

Define the abstract [Register](#) class.

Version:

Id

[class.register.php](#),v 1.1 2008-08-25 05:30:44 oscar Exp

7.14.2 Enumeration Type Documentation

7.14.2.1 enum OWL_APPLICATION_PATTERN

Bits 1-8 define the application. Application identifiers with the first bit set (0x40 - 0xff) are reserved for Oveas. 0xff is the [OWL](#) Identifier.

7.14.2.2 enum OWL_OBJECT_PATTERN

Bits 9-20 define the object type of an application

7.14.2.3 enum OWL_SEVERITY_PATTERN

Bits 29-32 define the severity

7.14.2.4 enum OWL_STATUS_PATTERN

Bits 21-28 defines the (object specific) status code

7.15 `/home/oscar/work/eclipse/owl-php/src/kernel/so/class.sessionhan` File Reference

Classes

- class [SessionHandler](#)
the PHP session object

7.15.1 Detailed Description

This file defines the [SessionHandler](#) class

Version:

Id

[class.sessionhandler.php](#),v 1.3 2008-09-02 05:16:53 oscar Exp

7.16 `/home/oscar/work/eclipse/owl-php/src/kernel/so/class.statushandler.php` File Reference

Classes

- class [StatusHandler](#)
Status object.

7.16.1 Detailed Description

This file defines status object that's user for all objects

Version:

Id

[class.statushandler.php](#), v 1.3 2008-09-02 05:16:53 oscar Exp

7.17 `/home/oscar/work/eclipse/owl-php/src/kernel/so/class.userhandle` File Reference

Classes

- class [UserHandler](#)
the user object

7.17.1 Detailed Description

This file defines the [UserHandler](#) class

Version:

Id

[class.userhandler.php](#),v 1.2 2008-09-02 05:16:53 oscar Exp

7.18 /home/oscar/work/eclipse/owl-php/src/kernel/ui/class.domelement.php File Reference

Classes

- class [DOMElement](#)
DOM Element base class.

7.18.1 Detailed Description

This file defines the top-level DOMElement class

Version:

Id

[class.domelement.php](#), v 1.1 2008-09-02 05:16:53 oscar Exp

7.19 `/home/oscar/work/eclipse/owl-php/src/kernel/ui/class.formelement` File Reference

Classes

- class [FormElement](#)
Form element.

7.19.1 Detailed Description

This file defines a form element

Version:

Id

[class.formelement.php](#), v 1.1 2008-09-02 05:16:53 oscar Exp

7.20 /home/oscar/work/eclipse/owl-php/src/lib/owl.messages.en-uk.php File Reference

Variables

- [\\$GLOBALS](#) ['messages']

7.20.1 Detailed Description

This file defines the message text for all status codes in UK English

Version:

Id

owl.messages.en-uk.php,v 1.4 2008-09-02 05:16:54 oscar Exp

7.20.2 Variable Documentation

7.20.2.1 \$GLOBALS['messages']

7.21 `/home/oscar/work/eclipse/owl-php/src/lib/owl.messages.php` File Reference

Variables

- `$GLOBALS` ['messages']

7.21.1 Detailed Description

This file defines the message text for all status codes in UK English

Version:

Id

`owl.messages.php`,v 1.3 2008-09-02 05:16:54 oscar Exp

7.21.2 Variable Documentation

7.21.2.1 `$GLOBALS['messages']`

7.22 /home/oscar/work/eclipse/owl-php/src/lib/owl.severitycodes.php File Reference

Enumerations

Severity codes

Define the Severity values for all status codes as returned by `_OWL::severity()`.

- enum [OWL_DEBUG](#)
- enum [OWL_INFO](#)
- enum [OWL_OK](#)
- enum [OWL_SUCCESS](#)
- enum [OWL_WARNING](#)
- enum [OWL_BUG](#)
- enum [OWL_ERROR](#)
- enum [OWL_FATAL](#)
- enum [OWL_CRITICAL](#)

7.22.1 Detailed Description

This file defines the Severity codes that all objects can have. It must be the very first file that is included, since these codes are used from step 1.

Version:

Id

[owl.severitycodes.php](#),v 1.2 2008-08-22 12:02:13 oscar Exp

7.22.2 Enumeration Type Documentation

7.22.2.1 enum OWL_BUG

Something weird happened, this might be a bug

7.22.2.2 enum OWL_CRITICAL

Pack your backs and start running. This status is reserved

7.22.2.3 enum OWL_DEBUG

Status that can be logged in debug mode

7.22.2.4 enum OWL_ERROR

Last operational ended with an error; current object cannot be trusted anymore

7.22.2.5 enum OWL_FATAL

Last operation had a fatal status; [OWL](#) environment cannot be trusted anymore

7.22.2.6 enum OWL_INFO

Holds neutral informarion about the current object status

7.22.2.7 enum OWL_OK

General normal status

7.22.2.8 enum OWL_SUCCESS

Last operation ended successful

7.22.2.9 enum OWL_WARNING

Last operation ended with a warning; this might be a temporataty status (e.g. network issue) or
a. user error

7.23 /home/oscar/work/eclipse/owl-php/src/OWLloader.php File Reference

Enumerations

- enum [OWL_INCLUDE](#)
- enum [OWL_SO_INC](#)
- enum [OWL_BO_INC](#)
- enum [OWL_UI_INC](#)
- enum [OWL_LIBRARY](#)

Variables

- [\\$GLOBALS](#) ['owl_object'] = new [OWL](#)()
- [\\$GLOBALS](#) ['messages'] = array ()
- [\\$GLOBALS](#) ['db']
- [\\$GLOBALS](#) ['formdata'] = & new [FormHandler](#)()
- [\\$GLOBALS](#) ['user'] = & new [User](#)()

7.23.1 Detailed Description

This file loads the [OWL](#) environment in intialises some singletons

Version:

Id

[OWLloader.php](#),v 1.4 2008-09-02 05:16:53 oscar Exp

7.23.2 Enumeration Type Documentation

7.23.2.1 enum [OWL_BO_INC](#)

7.23.2.2 enum [OWL_INCLUDE](#)

7.23.2.3 enum [OWL_LIBRARY](#)

7.23.2.4 `enum OWL_SO_INC`

7.23.2.5 `enum OWL_UI_INC`

7.23.3 Variable Documentation

7.23.3.1 `$GLOBALS['user'] = & new User()`

7.23.3.2 `$GLOBALS['formdata'] = & new FormHandler()`

7.23.3.3 `$GLOBALS['db']`

Initial value:

```
& new DBHandler(  
    ConfigHandler::get ('dbserver')  
    , ConfigHandler::get ('dbname')  
    , ConfigHandler::get ('dbuser')  
    , ConfigHandler::get ('dbpasswd'))
```

7.23.3.4 `$GLOBALS['messages'] = array ()`

7.23.3.5 `$GLOBALS['owl_object'] = new OWL()`

7.24 /home/oscar/work/eclipse/owl-php/src/OWLrundown.php File Reference

7.24.1 Detailed Description

Make sure all objects are destroyed in the proper order

Version:

Id

[OWLrundown.php](#),v 1.2 2008-09-02 05:16:53 oscar Exp

Index

- [/home/oscar/work/eclipse/owl-php/src/OWLloader.php, 107](#)
- [/home/oscar/work/eclipse/owl-php/src/OWLrundown.php, 109](#)
- [/home/oscar/work/eclipse/owl-php/src/config.php, 79](#)
- [/home/oscar/work/eclipse/owl-php/src/index.php, 81](#)
- [/home/oscar/work/eclipse/owl-php/src/kernel/bo/class.form.php, 82](#)
- [/home/oscar/work/eclipse/owl-php/src/kernel/bo/class.session.php, 83](#)
- [/home/oscar/work/eclipse/owl-php/src/kernel/bo/class.user.php, 84](#)
- [/home/oscar/work/eclipse/owl-php/src/kernel/class._OWL.php, 85](#)
- [/home/oscar/work/eclipse/owl-php/src/kernel/so/class.confighandler.php, 86](#)
- [/home/oscar/work/eclipse/owl-php/src/kernel/so/class.datahandler.php, 87](#)
- [/home/oscar/work/eclipse/owl-php/src/kernel/so/class.dbhandler.php, 90](#)
- [/home/oscar/work/eclipse/owl-php/src/kernel/so/class.exceptionhandler.php, 92](#)
- [/home/oscar/work/eclipse/owl-php/src/kernel/so/class.filehandler.php, 93](#)
- [/home/oscar/work/eclipse/owl-php/src/kernel/so/class.formhandler.php, 94](#)
- [/home/oscar/work/eclipse/owl-php/src/kernel/so/class.loghandler.php, 95](#)
- [/home/oscar/work/eclipse/owl-php/src/kernel/so/class.register.php, 96](#)
- [/home/oscar/work/eclipse/owl-php/src/kernel/so/class.sessionhandler.php, 98](#)
- [/home/oscar/work/eclipse/owl-php/src/kernel/so/class.statushandler.php, 99](#)
- [/home/oscar/work/eclipse/owl-php/src/kernel/so/class.userhandler.php, 100](#)
- [/home/oscar/work/eclipse/owl-php/src/kernel/ui/class.domelement.php, 101](#)
- [/home/oscar/work/eclipse/owl-php/src/kernel/ui/class.formelement.php, 102](#)
- [/home/oscar/work/eclipse/owl-php/src/lib/owl.messages.en-uk.php, 103](#)
- [/home/oscar/work/eclipse/owl-php/src/lib/owl.messages.php, 104](#)
- [/home/oscar/work/eclipse/owl-php/src/lib/owl.severitycodes.php, 105](#)
- [\\$GLOBALS
 - \[config.php, 79, 80\]\(#\)
 - \[owl.messages.en-uk.php, 103\]\(#\)
 - \[owl.messages.php, 104\]\(#\)
 - \[OWLloader.php, 108\]\(#\)](#)
- [\\$caller
 - \[OWLErrorException, 51\]\(#\)](#)
- [\\$code
 - \[StatusHandler, 68\]\(#\)](#)
- [\\$database
 - \[DbHandler, 30\]\(#\)](#)
- [\\$dataset
 - \[Session, 60\]\(#\)
 - \[SessionHandler, 65\]\(#\)
 - \[User, 73\]\(#\)
 - \[UserHandler, 78\]\(#\)](#)
- [\\$db_prefix
 - \[DbHandler, 30\]\(#\)](#)
- [\\$errno
 - \[DbHandler, 30\]\(#\)](#)
- [\\$error
 - \[DbHandler, 30\]\(#\)](#)

- \$filename
 - LogHandler, [46](#)
- \$fpointer
 - LogHandler, [46](#)
- \$id
 - DbHandler, [30](#)
- \$opened
 - DbHandler, [30](#)
 - LogHandler, [46](#)
- \$original_name
 - FileHandler, [34](#)
- \$owl_data
 - DataHandler, [22](#)
- \$owl_database
 - DataHandler, [23](#)
- \$owl_joins
 - DataHandler, [22](#)
- \$owl_keys
 - DataHandler, [23](#)
- \$owl_multivalues
 - FormHandler, [43](#)
- \$owl_prepared
 - DataHandler, [23](#)
- \$owl_tablename
 - DataHandler, [23](#)
- \$params
 - StatusHandler, [68](#)
- \$pstatus
 - _OWL, [15](#)
- \$query
 - DbHandler, [30](#)
- \$rowcount
 - DbHandler, [30](#)
- \$session
 - User, [73](#)
 - UserHandler, [78](#)
- \$severity
 - _OWL, [15](#)
- \$status
 - _OWL, [15](#)
- \$thrown_code
 - OWLException, [51](#)
- \$user_data
 - User, [73](#)
 - UserHandler, [78](#)
- _OWL, [13](#)
 - \$pstatus, [15](#)
 - \$severity, [15](#)
 - \$status, [15](#)
 - __destruct, [14](#)
 - init, [14](#)
 - reset, [14](#)
 - reset_calltree, [15](#)
- __construct
 - DataHandler, [20](#)
 - DbHandler, [26](#)
 - DOMElement, [32](#)
 - FileHandler, [33](#)
 - Form, [36](#)
 - FormElement, [38](#)
 - FormHandler, [41](#)
 - LogHandler, [45](#)
 - OWL, [49](#)
 - OWLException, [50](#)
 - Session, [58](#)
 - SessionHandler, [63](#)
 - StatusHandler, [66](#)
 - User, [71](#)
 - UserHandler, [76](#)
- __destruct
 - _OWL, [14](#)
 - DbHandler, [26](#)
 - FileHandler, [33](#)
 - LogHandler, [45](#)
 - Session, [58](#)
 - SessionHandler, [63](#)
 - User, [71](#)
 - UserHandler, [76](#)
- __get
 - DataHandler, [21](#)
 - FormHandler, [41](#)
- __set
 - DataHandler, [21](#)
 - FormHandler, [42](#)
- Business Object modules, [10](#)
- class.datahandler.php
 - DATA_DELETE, [88](#)
 - DATA_READ, [88](#)
 - DATA_RESET_FULL, [88](#)
 - DATA_RESET_META, [88](#)
 - DATA_RESET_PREPARE, [88](#)
 - DATA_RESET_STATUS, [88](#)
 - DATA_UNPREPARED, [88](#)
 - DATA_UPDATE, [89](#)
 - DATA_WRITE, [89](#)
- class.dbhandler.php
 - DBHANDLE_DATA, [90](#)
 - DBHANDLE_FIELDCOUNT, [91](#)
 - DBHANDLE_ROWCOUNT, [91](#)
 - DBHANDLE_SINGLEFIELD, [91](#)
 - DBHANDLE_SINGLEROW, [91](#)
 - DBHANDLE_TOTALFIELDCOUNT, [91](#)
- class.filehandler.php
 - FILE_NOTRIM, [93](#)
 - FILE_TRIM_C, [93](#)
 - FILE_TRIM_L, [93](#)

- FILE_TRIM_R, 93
- class.register.php
 - OWL_APPLICATION_PATTERN, 96
 - OWL_OBJECT_PATTERN, 96
 - OWL_SEVERITY_PATTERN, 96
 - OWL_STATUS_PATTERN, 96
- close
 - FileHandler, 34
 - Session, 58
 - SessionHandler, 63
- close_logfile
 - LogHandler, 46
- config.php
 - \$GLOBALS, 79, 80
- ConfigHandler, 16
 - convert, 16
 - get, 16
 - read_config, 16
- connect
 - DbHandler, 27
- convert
 - ConfigHandler, 16
- create
 - DbHandler, 27
- DATA_DELETE
 - class.datahandler.php, 88
- DATA_READ
 - class.datahandler.php, 88
- DATA_RESET_FULL
 - class.datahandler.php, 88
- DATA_RESET_META
 - class.datahandler.php, 88
- DATA_RESET_PREPARE
 - class.datahandler.php, 88
- DATA_RESET_STATUS
 - class.datahandler.php, 88
- DATA_UNPREPARED
 - class.datahandler.php, 88
- DATA_UPDATE
 - class.datahandler.php, 89
- DATA_WRITE
 - class.datahandler.php, 89
- DataHandler, 18
 - \$owl_data, 22
 - \$owl_database, 23
 - \$owl_joins, 22
 - \$owl_keys, 23
 - \$owl_prepared, 23
 - \$owl_tablename, 23
 - __construct, 20
 - __get, 21
 - __set, 21
 - find_field, 21
 - init, 22
 - reset, 20, 22
 - set_join, 22
 - set_key, 21
- DBHANDLE_DATA
 - class.dbhandler.php, 90
- DBHANDLE_FIELDCOUNT
 - class.dbhandler.php, 91
- DBHANDLE_ROWCOUNT
 - class.dbhandler.php, 91
- DBHANDLE_SINGLEFIELD
 - class.dbhandler.php, 91
- DBHANDLE_SINGLEROW
 - class.dbhandler.php, 91
- DBHANDLE_TOTALFIELDCOUNT
 - class.dbhandler.php, 91
- DbHandler, 24
 - \$database, 30
 - \$db_prefix, 30
 - \$errno, 30
 - \$error, 30
 - \$id, 30
 - \$opened, 30
 - \$query, 30
 - \$rowcount, 30
 - __construct, 26
 - __destruct, 26
 - connect, 27
 - create, 27
 - dbread, 29
 - escape_string, 29
 - init, 29
 - open, 27
 - read, 28
 - reset, 27
 - set_query, 28
 - table_description, 27
 - tablename, 28
 - unescape_string, 29
- dbread
 - DbHandler, 29
- destroy
 - Session, 59
 - SessionHandler, 64
- DOMElement, 31
 - __construct, 32
 - init, 32
 - reset, 32
- download
 - FileHandler, 34
- encode
 - FileHandler, 34
- escape_string

- DbHandler, [29](#)
- FILE_NOTRIM
 - class.filehandler.php, [93](#)
- FILE_TRIM_C
 - class.filehandler.php, [93](#)
- FILE_TRIM_L
 - class.filehandler.php, [93](#)
- FILE_TRIM_R
 - class.filehandler.php, [93](#)
- FileHandler, [33](#)
 - \$original_name, [34](#)
 - __construct, [33](#)
 - __destruct, [33](#)
 - close, [34](#)
 - download, [34](#)
 - encode, [34](#)
 - open, [33](#)
 - read_data, [34](#)
 - read_line, [34](#)
- find_field
 - DataHandler, [21](#)
- Form, [35](#)
 - __construct, [36](#)
 - init, [36](#)
 - reset, [36](#)
- FormElement, [37](#)
 - __construct, [38](#)
 - init, [38](#)
 - reset, [38](#)
- FormHandler, [39](#)
 - \$owl_multivalues, [43](#)
 - __construct, [41](#)
 - __get, [41](#)
 - __set, [42](#)
 - init, [43](#)
 - parse_formdata, [41](#)
 - reset, [42](#)
 - set_join, [42](#)
 - set_key, [42](#)
- gc
 - Session, [59](#)
 - SessionHandler, [64](#)
- get
 - ConfigHandler, [16](#)
- get_code
 - Register, [55](#)
 - StatusHandler, [67](#)
- get_message
 - StatusHandler, [68](#)
- get_run_id
 - Register, [55](#)
- get_severity
 - Register, [54](#)
 - StatusHandler, [67](#)
- get_severity_level
 - Register, [54](#)
- get_trace
 - OWLException, [51](#)
- get_username
 - User, [72](#)
 - UserHandler, [77](#)
- handle_exception
 - OWLExceptionHandler, [52](#)
- hash_password
 - UserHandler, [77](#)
- index.php
 - OWL_ROOT, [81](#)
- init
 - _OWL, [14](#)
 - DataHandler, [22](#)
 - DbHandler, [29](#)
 - DOMElement, [32](#)
 - Form, [36](#)
 - FormElement, [38](#)
 - FormHandler, [43](#)
 - LogHandler, [46](#)
 - OWL, [49](#)
 - Register, [53](#)
 - Session, [60](#)
 - SessionHandler, [64](#)
 - User, [72](#)
 - UserHandler, [77](#)
- Library (codes, messages files etc.), [12](#)
- log_exception
 - OWLExceptionHandler, [52](#)
- LogHandler, [44](#)
 - \$filename, [46](#)
 - \$fpointer, [46](#)
 - \$opened, [46](#)
 - __construct, [45](#)
 - __destruct, [45](#)
 - close_logfile, [46](#)
 - init, [46](#)
 - open_logfile, [46](#)
 - reset, [46](#)
 - set_filename, [46](#)
 - write_logfile, [46](#)
- login
 - User, [71](#), [72](#)
 - UserHandler, [76](#)
- logout
 - User, [72](#)
 - UserHandler, [76](#)

- open
 - DbHandler, [27](#)
 - FileHandler, [33](#)
 - Session, [58](#)
 - SessionHandler, [63](#)
- open_logfile
 - LogHandler, [46](#)
- OWL, [48](#)
 - __construct, [49](#)
 - init, [49](#)
 - reset, [49](#)
 - s, [49](#)
- owl.messages.en-uk.php
 - \$GLOBALS, [103](#)
- owl.messages.php
 - \$GLOBALS, [104](#)
- owl.severitycodes.php
 - OWL_BUG, [105](#)
 - OWL_CRITICAL, [105](#)
 - OWL_DEBUG, [105](#)
 - OWL_ERROR, [105](#)
 - OWL_FATAL, [106](#)
 - OWL_INFO, [106](#)
 - OWL_OK, [106](#)
 - OWL_SUCCESS, [106](#)
 - OWL_WARNING, [106](#)
- OWL_APPLICATION_PATTERN
 - class.register.php, [96](#)
- OWL_BO_INC
 - OWLloader.php, [107](#)
- OWL_BUG
 - owl.severitycodes.php, [105](#)
- OWL_CRITICAL
 - owl.severitycodes.php, [105](#)
- OWL_DEBUG
 - owl.severitycodes.php, [105](#)
- OWL_ERROR
 - owl.severitycodes.php, [105](#)
- OWL_FATAL
 - owl.severitycodes.php, [106](#)
- OWL_INCLUDE
 - OWLloader.php, [107](#)
- OWL_INFO
 - owl.severitycodes.php, [106](#)
- OWL_LIBRARY
 - OWLloader.php, [107](#)
- OWL_OBJECT_PATTERN
 - class.register.php, [96](#)
- OWL_OK
 - owl.severitycodes.php, [106](#)
- OWL_ROOT
 - index.php, [81](#)
- OWL_SEVERITY_PATTERN
 - class.register.php, [96](#)
- OWL_SO_INC
 - OWLloader.php, [107](#)
- OWL_STATUS_PATTERN
 - class.register.php, [96](#)
- OWL_SUCCESS
 - owl.severitycodes.php, [106](#)
- OWL_UI_INC
 - OWLloader.php, [108](#)
- OWL_WARNING
 - owl.severitycodes.php, [106](#)
- OWLException, [50](#)
 - \$caller, [51](#)
 - \$thrown_code, [51](#)
 - __construct, [50](#)
 - get_trace, [51](#)
 - stack_dump, [51](#)
 - traceback, [51](#)
- OWLExceptionHandler, [52](#)
 - handle_exception, [52](#)
 - log_exception, [52](#)
- OWLloader.php
 - \$GLOBALS, [108](#)
 - OWL_BO_INC, [107](#)
 - OWL_INCLUDE, [107](#)
 - OWL_LIBRARY, [107](#)
 - OWL_SO_INC, [107](#)
 - OWL_UI_INC, [108](#)
- parse_formdata
 - FormHandler, [41](#)
- Presentation modules, [9](#)
- read
 - DbHandler, [28](#)
 - Session, [59](#)
 - SessionHandler, [63](#)
- read_config
 - ConfigHandler, [16](#)
- read_data
 - FileHandler, [34](#)
- read_line
 - FileHandler, [34](#)
- read_userdata
 - UserHandler, [77](#)
- Register, [53](#)
 - get_code, [55](#)
 - get_run_id, [55](#)
 - get_severity, [54](#)
 - get_severity_level, [54](#)
 - init, [53](#)
 - register_app, [53](#)
 - register_class, [53](#)
 - register_code, [54](#)
 - register_severity, [54](#)

- set_application, 55
- set_class, 55
- set_severity, 55
- register_app
 - Register, 53
- register_class
 - Register, 53
- register_code
 - Register, 54
- register_severity
 - Register, 54
- reset
 - _OWL, 14
 - DataHandler, 20, 22
 - DbHandler, 27
 - DOMElement, 32
 - Form, 36
 - FormElement, 38
 - FormHandler, 42
 - LogHandler, 46
 - OWL, 49
 - Session, 60
 - SessionHandler, 64
 - StatusHandler, 67
 - User, 72
 - UserHandler, 77
- reset_calltree
 - _OWL, 15
- s
 - OWL, 49
- Session, 56
 - \$dataset, 60
 - __construct, 58
 - __destruct, 58
 - close, 58
 - destroy, 59
 - gc, 59
 - init, 60
 - open, 58
 - read, 59
 - reset, 60
 - write, 59
- SessionHandler, 61
 - \$dataset, 65
 - __construct, 63
 - __destruct, 63
 - close, 63
 - destroy, 64
 - gc, 64
 - init, 64
 - open, 63
 - read, 63
 - reset, 64
 - write, 64
- set_application
 - Register, 55
- set_class
 - Register, 55
- set_code
 - StatusHandler, 67
- set_filename
 - LogHandler, 46
- set_join
 - DataHandler, 22
 - FormHandler, 42
- set_key
 - DataHandler, 21
 - FormHandler, 42
- set_params
 - StatusHandler, 67
- set_query
 - DbHandler, 28
- set_severity
 - Register, 55
- set_username
 - User, 72
 - UserHandler, 77
- stack_dump
 - OWLException, 51
- StatusHandler, 66
 - \$code, 68
 - \$params, 68
 - __construct, 66
 - get_code, 67
 - get_message, 68
 - get_severity, 67
 - reset, 67
 - set_code, 67
 - set_params, 67
- Storage Object modules, 11
- table_description
 - DbHandler, 27
- tablename
 - DbHandler, 28
- traceback
 - OWLException, 51
- unescape_string
 - DbHandler, 29
- User, 69
 - \$dataset, 73
 - \$session, 73
 - \$user_data, 73
 - __construct, 71
 - __destruct, 71
 - get_username, 72

- init, [72](#)
- login, [71](#), [72](#)
- logout, [72](#)
- reset, [72](#)
- set_username, [72](#)
- UserHandler, [74](#)
 - \$dataset, [78](#)
 - \$session, [78](#)
 - \$user_data, [78](#)
 - __construct, [76](#)
 - __destruct, [76](#)
 - get_username, [77](#)
 - hash_password, [77](#)
 - init, [77](#)
 - login, [76](#)
 - logout, [76](#)
 - read_userdata, [77](#)
 - reset, [77](#)
 - set_username, [77](#)
- write
 - Session, [59](#)
 - SessionHandler, [64](#)
- write_logfile
 - LogHandler, [46](#)