

DBeasy



Contents

	2
choose_tab_field.php	3
choose_tab_field_exec.php	4
classes.php	5
access_connection_form.php	6
odbc_connection_form.php	7
sqlite_connection_form.php	8
standard_connection_form.php	9
connection_error.php	10
create_textfield.php	11
create_textfield_process.php	12
database_show_table.php	13
dbeasy_security.php	14
Function inputcheck	14
filefunctions.php	15
Function loadHelpFile	15
Function my_copy	15
Function my_dir_copy	15
FileGenerator.php	16
formular_add_choose_exec.php	17
formular_add_choose_formular.php	18
formular_add_choose_project.php	19
formular_create_check.php	20
formular_create_exec.php	21
formular_create_specify.php	22
formular_load.php	23
form_dependency.php	24
form_dependency_exec.php	25
functions.php	26
Function translateFormType	26
functions_advanced.php	27
Function arrayToStrings	27
Function clearDependency	27
Function getDependency	27
Function getFieldsForTable	27
Function getFieldType	28
Function getFormsByType	28
Function getFormsWithDependency	28
Function getMenu	28
Function getPossibleAppearance	29
Function getPossibleTablesAfterJoin	29
Function makeMenu2	29
Function makeMenuMultyArray	30

Function makeMenuMultyArray2	30
Function primaryIntKeyExists	30
Function removeDBeaslyTablesFromArray	31
Function removeEntryFromArray	31
functions copy_form.php	32
Function checkNameExists	32
Function copyFormToForm	32
Function copyFormToReport	32
Function copyReportToForm	33
functions help.php	34
Function showHelp	34
general functions.php	35
Function arrayToString	35
Function getFormType	35
Function previewLoad	35
Function reloadPreview	35
headline_exec.php	37
headline_specify.php	38
help_window.php	39
index.php	40
index_create.php	41
inputtochange.php	42
inputtochange_exec.php	43
install_db_connection.php	44
install_db_exec.php	45
install_db_specify.php	46
join_delete.php	47
join_exec.php	48
join_exec_single.php	49
join_specify.php	50
join_specify_search.php	51
layout_exec.php	52
layout_report_exec.php	53
layout_report_specify.php	54
Function checkLayout	54
layout_specify.php	55
login.php	56
login_exec.php	57
logo_exec.php	58
logo_specify.php	59
mainmenu formularadmin.php	60
mainmenu sysadmin.php	62
menu formularadmin formular.php	63
menu formularadmin formular aenderung.php	64
menu formularadmin formular bericht.php	65
menu formularadmin formular eintrag.php	66
menu formularadmin formular suche.php	67
menu formularadmin project.php	68
menu sysadmin manage formularadmin.php	69

menu_sysadmin_manage_sysadmin.php	70
preview_aenderung.php	71
preview_eintrag.php	72
preview_report.php	73
Function reportIsEmpty	73
preview_suche.php	74
project_create_exec.php	75
project_create_specify.php	76
project_delete.php	77
project_delete_form.php	78
project_delete_warn.php	79
project_edit.php	80
project_show_formular.php	81
publish_aenderung.php	82
publish_aenderung_exec.php	83
publish_css.php	84
Function generateCSSFilename	84
Function generateForms	84
Function generatePreviewForms	84
Function generateStyles	84
Function getFormStyle	84
Function getSectionStyle	84
Function getStyleName	84
Function publishCSS	84
Function publishFormCSS	84
Function publishReportCSS	84
publish_eintrag.php	85
publish_eintrag_exec.php	86
publish_report.php	87
publish_report_exec.php	88
publish_search.php	89
publish_search_exec.php	90
reportfield_exec.php	91
report_exec.php	92
report_fieldspecify.php	93
report_functions.php	94
Function allfields	94
Function makeHTML	94
Function publishForm	94
Function searchKeyInArray	95
Function write1DimStringArray	95
Function write2DimStringArray	95
report_specify.php	96
search_exec.php	97
section_management_exec.php	98
Function arrayToString2	98
section_management_specify.php	99
Function table_tr	99
security_check_logged_in.php	100

security_functions.php	101
Function checkForm	101
Function compareFields	101
Function compareTables	101
Function fieldError	101
Function getTablesAndFields	101
Function push_error	101
Function tableError	101
SQLFunctions.php	102
Define ADODB_FORCE_NULLS	102
SQLInstallFunctions.php	103
start.php	104
style_editor.php	105
Function arrayAdjustLength	105
Function ifEqual	105
style_editor_process.php	106
Function makeInsertArray	106
style_editor_update.php	107
style_index.php	108
Function getStyleNames	108
style_index_process_1.php	109
style_index_process_2.php	110
style_index_process_3.php	111
table_add_field_exec.php	112
table_add_field_specify.php	113
table_add_field_warn.php	114
table_create.php	115
table_create_exec.php	116
table_create_specify.php	117
table_drop.php	118
table_drop_column.php	119
table_drop_column_warn.php	120
table_drop_warn.php	121
table_show_structure.php	122
template.php	123
uninstall.php	124
uninstall_exec.php	125
user_db_mapping.php	126
user_db_mapping_connection.php	127
user_db_mapping_delete.php	128
user_db_mapping_exec.php	129
user_db_mapping_specify.php	130
user_db_mapping_warn.php	131
user_delete.php	132
user_delete_warn.php	133
user_dir_mapping.php	134
user_dir_mapping_delete.php	135
user_dir_mapping_exec.php	136
user_dir_mapping_warn.php	137

user_entry_exec.php	138
user_entry_specify.php	139
user_show_table.php	140
Package DBeasy Classes	141
Class Error	141
Var \$filename	141
Constructor Error	141
Method addError	141
Method deleteFile	141
Method getErrors	141
Class FileGenerator	142
Var \$file_pointer	142
Method closeFile	142
Method createFile	142
Method my_split	142
Method write	142
Class fileinput	143
Var \$files	143
Constructor fileinput	143
Method close	143
Method read	143
Method readAll	143
Class fileoutput	144
Var \$files	144
Constructor fileoutput	144
Method close	144
Method write	144
Method writeln	144
Class Formular	145
Var \$debug	145
Var \$dependency	145
Var \$headline	145
Var \$id	145
Var \$layout	145
Var \$logic	145
Var \$logo	145
Var \$name	146
Var \$project_id	146
Var \$sections	146
Var \$style	146
Var \$type	146
Method addEnumeration	146
Method addFormField	146
Method addTextField	146
Method create	147
Method deleteForm	147
Method deleteSection	147
Method existsSection	147
Method formular	148

Method <u>getSection</u>	148
Method <u>getSections</u>	148
Method <u>getValues</u>	148
Method <u>load</u>	148
Method <u>moveSectionDown</u>	148
Method <u>moveSectionUp</u>	148
Method <u>setDependency</u>	149
Method <u>setEnumeration</u>	149
Method <u>setFormField</u>	149
Method <u>setHeadline</u>	150
Method <u>setLayout</u>	150
Method <u>setLogo</u>	150
Method <u>setName</u>	150
Method <u>setStyle</u>	150
Method <u>setTextField</u>	151
Method <u>updateDB</u>	151
Class <u>HTMLGenerator</u>	151
Method <u>makeBR</u>	151
Method <u>makeChangePrimaryKeyLink</u>	151
Method <u>makeCheckbox</u>	152
Method <u>makeDropDownMenu</u>	152
Method <u>makeEasyDropDownMenu</u>	152
Method <u>makeFormBegin</u>	152
Method <u>makeFormEnd</u>	153
Method <u>makeFormGetBegin</u>	153
Method <u>makeFormHead</u>	153
Method <u>makeFormularButton</u>	153
Method <u>makeHTMLEnd</u>	153
Method <u>makeHTMLHead</u>	154
Method <u>makeLabel</u>	154
Method <u>makeMultiLineInputField</u>	154
Method <u>makeRadioButton</u>	154
Method <u>makeReturnToIndexLink</u>	155
Method <u>makeSingleLineInputField</u>	155
Method <u>makeTableBegin</u>	155
Method <u>makeTableEnd</u>	155
Method <u>makeTDBegin</u>	155
Method <u>makeTDColspanBegin</u>	155
Method <u>makeTDEnd</u>	156
Method <u>makeTextField</u>	156
Method <u>makeTRBegin</u>	156
Method <u>makeTREnd</u>	156
Method <u>makeULLink</u>	156
Method <u>transmitChangedPrimaryKeyValue</u>	156
Method <u>writeln</u>	156
Class <u>PHPGenerator</u>	157
Method <u>makeAutoReturn</u>	157
Method <u>makeChangeIDFormular</u>	157
Method <u>makeConnection</u>	157

Method makeFormHead	158
Method makeHTMLHead	158
Method makeIncludeAdo	158
Method makeInputCheck	158
Method makePHPBegin	159
Method makePHPEnd	159
Method makeReturnButton	159
Method makeReturnToIndexLink	159
Method makeSQL_insert	159
Method makeSQL_update	159
Class Section	160
 Var \$appearance	160
 Var \$data_type	160
 Var \$debug	160
 Var \$enumerations	160
 Var \$field	160
 Var \$form_id	160
 Var \$id	160
 Var \$label	160
 Var \$my_sql	161
 Var \$position	161
 Var \$table	161
 Var \$text	161
 Constructor Section	161
 Method addEnumeration	161
 Method createSection	161
 Method deleteEnumeration	161
 Method deleteSection	161
 Method getValues	162
 Method isText	162
 Method loadEnumerations	162
 Method loadSection	162
 Method setFormField	162
 Method setPosition	162
 Method setTextField	162
 Method updateDB	163
Class SQLFunctions	163
 Var \$connection	163
 Var \$current_result	163
 Var \$db_address	163
 Var \$db_dsn	163
 Var \$db_name	163
 Var \$db_password	163
 Var \$db_path	163
 Var \$db_type	163
 Var \$db_username	164
 Var \$debug	164
 Constructor SQLFunctions	164
 Method addField	164

Method addUser	164
Method autoConnectToDB	164
Method connectToDB	164
Method connectToDBODBM	165
Method createTable	165
Method deleteRow	165
Method dropColumn	166
Method dropTable	166
Method executeSQL	166
Method existsItem	166
Method generateDate	166
Method generateID	166
Method getFields	167
Method getIntPrimaryKey	167
Method getMax	167
Method getResultMsg	167
Method insertData	167
Method LogTable	168
Method selectAll	168
Method showTables	168
Method tableExists	168
Method updateRow	169
Class SQLInstallFunctions	169
Method createConditionTable	169
Method createFormularTable	170
Method createHelpTable	170
Method createJoinTable	170
Method createLogicTable	170
Method createLogTable	170
Method createProjectTable	171
Method createReportCondTable	171
Method createReportFieldTable	171
Method createSectionTable	171
Method createStyleFormsTable	171
Method createStyleSectionsTable	172
Method createStyleStylesTable	172
Method createUserDBDataFile	172
Method createUserDBMappingTable	172
Method createUserDirMappingTable	173
Method createUserTable	173
Method createWorkfieldTable	173
Appendices	174
Appendix A - Class Trees	175
DBeasy	175

Package DBeasy

```
<table border="0" align="center" height="100%">
<tr>
  <td calign="middle" align="center" class="main">
    <div align="center">
      <a href="http://sourceforge.net/projects/dbeasy/" target="_blank"></a>
    <br>
```

FitH

This page appears if the user has uninstalled dbeasy correctly.

- **Package DBeasy**

choose_tab_field.php

Selecting an formfield with all necessary values:

Table, Field, Label, Appearance and if necessary default values for drop-down menus or radiobuttons

- **Package** DBeasy

include_once ["functions_help.php"](#)*[line 25]*

include_once ["functions_advanced.php"](#)*[line 10]*

include_once ["classes.php"](#)*[line 9]*

choose_tab_field_exec.php

Here the actual formfield section is created and added to the database

- **Package** DBeasy

include_once ["functions_advanced.php"](#)*[line 21]*

include_once ["general_functions.php"](#)*[line 20]*

include_once ["classes.php"](#)*[line 6]*

classes.php

Core classes (Section and Formular are defined here)

- **Package** DBeasy

include_once ["filefunctions.php"](#)*[line 12]*

include_once ["general_functions.php"](#)*[line 11]*

include_once ["SQLFunctions.php"](#)*[line 10]*

include_once ["security_check_logged_in.php"](#)*[line 9]*

access_connection_form.php

This file displays input fields required to connect to access

- **Package** DBeasy

odbc_connection_form.php

This file displays input fields required to connect to odbc

- **Package** DBeasy

sqlite_connection_form.php

This file displays input fields required to connect to sqlite

- **Package** DBeasy

standard_connection_form.php

This file displays input fields required to connect to standard databases like mysql

- **Package** DBeasy

connection_error.php

This file appears if there is a problem with connecting to the database

- **Package** DBeasy

create_textfield.php

This file provides the possibility to create a textfield in a formular.

Textfields contain nothing, but text that will NOT be stored in the target database. An example use of textfields are explanations/comments concerning some formular fields.

- **Package** DBeasy

include ["functions_help.php"](#)*[line 21]*

create_textfield_process.php

This file performs the creation of a textfield in a formular.

- **Package** DBeasy

include_once ["general_functions.php"](#)*[line 7]*

include_once ["classes.php"](#)*[line 6]*

database_show_table.php

This file provides an overview of the tables within a selected database.

- **Package** DBeasy

include ["functions_help.php"](#)*[line 21]*

include_once ["SQLFunctions.php"](#)*[line 9]*

dbeasy_security.php

This file contains some functions for SQL injection security.

- **Package** DBeasy

function inputcheck(\$return_link) [*line 11*]

Function Parameters:

- *string* **\$return_link** Page where to return if failed

Checks on SQL Injection attacks (;, ', --)

filefunctions.php

This class provides functions to deal with files like reading or writing

- **Package** DBeasy

function loadHelpFile() [*line 237*]

function my_copy(\$oldname, \$newname) [*line 186*]

Function Parameters:

- *string* **\$oldname** Source
- *string* **\$newname** Destination

Copies files from a path to another path

bool function my_dir_copy(\$oldname, \$newname) [*line 209*]

Function Parameters:

- *string* **\$oldname** Source
- *string* **\$newname** Destination

Copies entire directories

FileGenerator.php

This file contains the classes: FileGenerator, HTMLGenerator and PHPGenerator.

- **Package** DBeasy

include_once ["SQLFunctions.php"](#)*[line 6]*

formular_add_choose_exec.php

This file performs the copying of a formular.

There are checks on existence and forbidden names of formulars.

- **Package** DBeasy

include_once ["FileGenerator.php"](#)*[line 9]*

include_once ["SQLFunctions.php"](#)*[line 8]*

formular_add_choose_formular.php

This file asks for the concrete formular that is going to be copied.

- **Package** DBeasy

include ["functions_help.php"](#)*[line 26]*

include_once ["SQLFunctions.php"](#)*[line 8]*

formular_add_choose_project.php

This file asks for the source of a formular during copying of formulars.

- **Package** DBeasy

include ["functions_help.php"](#)*[line 21]*

include_once ["SQLFunctions.php"](#)*[line 7]*

formular_create_check.php

This file checks for existing and forbidden names of a formular.

- **Package** DBeasy

include_once ["general_functions.php"](#)*[line 46]*

include_once ["SQLFunctions.php"](#)*[line 45]*

formular_create_exec.php

This file performs the creation of a new formular..

- **Package** DBeasy

include_once ["classes.php"](#)*[line 7]*

formular_create_specify.php

This file provides a HTML formular for creation of a new formular.

Below the HTML formular existing formulars are enumerated.

- **Package** DBeasy

include ["functions_help.php"](#)*[line 57]*

include_once ["SQLFunctions.php"](#)*[line 8]*

formular_load.php

This file loads a formular specified with its ID

- **Package** DBeasy

include ["classes.php"](#)*[line 10]*

form_dependency.php

Here the dependencies between search - report - change formulars can be set

- **Package** DBeasy

include_once ["functions_help.php"](#)*[line 11]*

include_once ["functions_advanced.php"](#)*[line 7]*

include_once ["classes.php"](#)*[line 6]*

form_dependency_exec.php

Here just the selected form dependencies are stored in the DB

- **Package** DBeasy

include_once ["classes.php"](#)*[line 7]*

functions.php

Collection of frequently used Functions

- **Package** DBeasy

function translateFormType(\$type) *[line 6]*

functions_advanced.php

A collection of functions that make use of the Database

- **Package** DBeasy

function arrayToStrings(\$values) [*line 46*]

Function Parameters:

- *array* **\$values** values array to convert

Converts an array into a string with ; as a separator

function clearDependency(\$dependency, \$type) [*line 361*]

Function Parameters:

- *int* **\$dependency** dependency Dependency of the form that we wanna lookup
- *string* **\$type** type Formtype of the form we wanna lookup

Clears the dependency of forms that have a given type and dependency

function getDependency(\$id) [*line 381*]

Function Parameters:

- *int* **\$id** id ID of the form from which we want to get the dependency

Returns the dependency of a form

function getFieldsForTable(\$tab, \$selected, [\$remove_primary = true]) [*line 118*]

Function Parameters:

- *string* **\$tab** tab Table the table from which we select the fieldnames

- *string* **\$selected** selected the value that should be selected in the menu
- *bool* **\$remove_primary** remove_primary whether to remove the primary integer keys

Creates an drop down menu content with all fieldnames from a table

function getFieldTypes(\$table, \$field) [*line 149*]

Function Parameters:

- *string* **\$table** table Table the table in which we wanna lookup the
- *string* **\$field** field Field which we wanna lookup

Returns the type of the field form the target database

function getFormsByType(\$type) [*line 281*]

Function Parameters:

- *string* **\$type** type Formtype we want to lookup

Returns all forms of a given Type

function getFormsWithDependency(\$dependency, \$type) [*line 334*]

Function Parameters:

- *int* **\$dependency** dependency Dependency of the form that we wanna lookup
- *string* **\$type** type Formtype of the form we wanna lookup

Returns forms of a given type whith a given dependency

function getMenu(\$tab, \$condition, \$value, \$menuname, \$column_show, \$column_submit) [*line 70*]

Function Parameters:

- *string* **\$tab** tab Table to select from

- *string* **\$condition** condition field which should have a certain value
- *string* **\$value** value the value we question
- *string* **\$menuname** menuname name of the menu
- *int* **\$column_show** column_show Column that should be displayed
- *int* **\$column_submit** column_submit Column that should be submitted

Creates an HTML menu from a database query

function getPossibleAppearance(\$field_type) [*line 176*]

Function Parameters:

- *string* **\$field_type** field_type an ADODB fieldtype

Returns the possible appearances array for a given fieldtype

function getPossibleTablesAfterJoin(\$form_id) [*line 93*]

Function Parameters:

- *int* **\$form_id** form_id form we are working on

Returns tables that are joined in the form

include_once ["SQLFunctions.php"](#) [*line 8*]

function makeMenu2(\$menuname, \$should_submit, \$values, \$selected_value) [*line 306*]

Function Parameters:

- *string* **\$menuname** menuname Name of the menu
- *bool* **\$should_submit** should_submit If the menu should be submitted on change
- *stringarray* **\$values** values array with the values
- *string* **\$selected_value** selected_value Value that should be selected

Displays an entire drop down menu built from a values array The displayed value will be submitted

function makeMenuMultyArray(\$menuname, \$should_submit, \$values, \$value, \$name) [*line 230*]

Function Parameters:

- *string* **\$menuname** menuname Name of the menu
- *boolean* **\$should_submit** should_submit if the menu should be submitted on change
- *string* **\$values** values the multidimensional array with the values
- *int* **\$value** value Column with the values that should be submitted
- *int* **\$name** name Column with the values that should be displayed

Displays and drop down menu based on an multy array

function makeMenuMultyArray2(\$menuname, \$should_submit, \$values, \$value, \$name, \$selected_value, [\$formname = "form"]) [*line 256*]

Function Parameters:

- *string* **\$menuname** menuname Name of the menu
- *boolean* **\$should_submit** should_submit if the menu should be submitted on change
- *string* **\$values** values the multidimensional array with the values
- *int* **\$value** value Column with the values that should be submitted
- *int* **\$name** name Column with the values that should be displayed
- *string* **\$selected_value** name selected_value the value that should be selected

Displays and drop down menu based on an multy array with additional parameters

function primaryIntKeyExists(\$tables) [*line 393*]

Function Parameters:

- *stringarray* **\$tables** tables tables we wanna get the integer primary key from

Returns the primary Integer key of \$table - or false if none exists

function removeDBeasyTablesFromArray(\$arr) *[line 32]*

Function Parameters:

- *stringarray* **\$arr** arr array from which we want to remove the DBEasy Sytem table names

Removes the dbeasy-tables from the given array and returns the reduced array

function removeEntryFromArray(\$arr, \$entry) *[line 15]*

Function Parameters:

- *array* **\$arr** arr an arbitrary array
- *string* **\$entry** entry an entry to this array

Removes an entry from a given array and returns this reduced array

functions_copy_form.php

Function to create a form out of another form

- **Package** DBeasy

function checkNameExists(\$name) [*line 12*]

function copyFormToForm(\$from_form, \$new_type, \$change_name, \$change_logo, \$change_headline, [\$change_fields = array()]) [*line 45*]

Function Parameters:

- *int* **\$from_form** from_form ID of the form we copy the values from
- *string* **\$new_type** new_type Type of the new form
- *string* **\$change_name** change_name Name of the new form
- *string* **\$change_logo** change_logo Path to the logo of the new form
- *string* **\$change_headline** change_headline Headline of the new form
- *array* **\$change_fields** change_fields The formfields we copy to the new form

Copies non reports into non reports

function copyFormToReport(\$from_form, \$new_type, \$change_name, \$change_logo, \$change_headline, [\$change_fields = array()]) [*line 93*]

Function Parameters:

- *int* **\$from_form** from_form ID of the form we copy the values from
- *string* **\$new_type** new_type Type of the new form
- *string* **\$change_name** change_name Name of the new form
- *string* **\$change_logo** change_logo Path to the logo of the new form
- *string* **\$change_headline** change_headline Headline of the new form
- *array* **\$change_fields** change_fields The formfields we copy to the new form

Copies non reports into reports

```
function copyReportToForm($from_form, $new_type, $change_name, $change_logo, $change_headline,
[$change_fields = array()]) [line 157]
```

Function Parameters:

- *int* **\$from_form** from_form ID of the form we copy the values from
- *string* **\$new_type** new_type Type of the new form
- *string* **\$change_name** change_name Name of the new form
- *string* **\$change_logo** change_logo Path to the logo of the new form
- *string* **\$change_headline** change_headline Headline of the new form
- *array* **\$change_fields** change_fields The formfields we copy to the new form

Copies reports into reports

```
include_once "classes.php" [line 7]
```

```
include_once "security\_functions.php" [line 9]
```

```
include_once "report\_functions.php" [line 8]
```

functions_help.php

This file contains functions necessary for the context sensitive help

- **Package** DBeasy

function showHelp(\$category) [*line 13*]

Function Parameters:

- *string* **\$category** category Category for which the help should be displayed

Shows the help icon with a link that pops up the help text of a category

general_functions.php

A collection of functions that don't need anything special

- **Package** DBeasy

function arrayToString(\$new_array, \$separator, \$last_seperator) [*line 14*]

Function Parameters:

- *array* **\$new_array** new_array Array to be converted
- *string* **\$separator** seperator seperator to be used
- *bool* **\$last_seperator** last_seperator if there should a seperated as last char

Returns a string built from an array

function getFormType(\$type) [*line 106*]

Function Parameters:

- *string* **\$type** type Formtype

Returns the formtype as an actual word

function previewLoad(\$mainframe_url) [*line 43*]

Function Parameters:

- *string* **\$mainframe_url** mainframe_url URL to be loaded

Loads the given url to the main frame and updates the preview if it is activated

function reloadPreview() [*line 74*]

Updates preview if activated

headline_exec.php

This file performs the setting of a headline for a formular.

- **Package** DBeasy

include ["classes.php"](#)*[line 6]*

include_once ["security_functions.php"](#)*[line 21]*

headline_specify.php

This file provides a HTML input field for the headline of a formular.

- **Package** DBeasy

include ["functions_help.php"](#)*[line 19]*

include ["classes.php"](#)*[line 6]*

help_window.php

This page enables the help popup with helptext

- **Package** DBeasy

include ["SQLFunctions.php"](#)*[line 18]*

index.php

This file redirects to the login interface.

- **Package** DBeasy

index_create.php

This file creates a project index.

- **Package** DBeasy

include **\$complete_path** [*line 95*]

include_once ["FileGenerator.php"](#) [*line 7*]

inputtochange.php

The values from the current form will be copied into a new form of the current project

Several values can be changed including name (necessary), headline, logo and formfields

- **Package** DBeasy

include ["functions_help.php"](#)*[line 28]*

include ["classes.php"](#)*[line 8]*

inputtochange_exec.php

Actually creates the new form from the current one based on the changes selected before

- **Package** DBeasy

include_once ["functions_copy_form.php"](#)*[line 7]*

include_once ["classes.php"](#)*[line 6]*

install_db_connection.php

This file provides an interface for inserting the access data for the DBeasy system database depending on the database type the user has chosen before.

- **Package DBeasy**

include ["connection/standard_connection_form.php"](#) [file 46]

include ["connection/standard_connection_form.php"](#) [file 44]

include ["connection/access_connection_form.php"](#) [file 42]

include ["connection/sqlite_connection_form.php"](#) [file 40]

include ["connection/odbc_connection_form.php"](#) [file 38]

install_db_exec.php

This file performs the installation and configuration of DBeasy and creating the first sysadmin account.

- **Package** DBeasy

include ["SQLInstallFunctions.php"](#)*[line 25]*

install_db_specify.php

This file provides an interface for installation and configuration of DBeasy and creating the first sysadmin account.

- **Package** DBeasy

join_delete.php

This file is responsible for deleting a join

- **Package** DBeasy

include ["classes.php"](#)*[line 9]*

include_once ["SQLFunctions.php"](#)*[line 8]*

include_once ["functions_advanced.php"](#)*[line 7]*

join_exec.php

This file is responsible for inserting a join

- **Package** DBeasy

include ["classes.php"](#)*[line 8]*

include_once ["functions_advanced.php"](#)*[line 9]*

join_exec_single.php

This file creates a join within a table

- **Package** DBeasy

include ["classes.php"](#)*[line 7]*

include_once ["functions_advanced.php"](#)*[line 8]*

join_specify.php

- **Package** DBeasy

include ["functions_help.php"](#)*[line 98]*

include ["classes.php"](#)*[line 2]*

join_specify_search.php

Just a redirection to the right place for setting Join Conditions

- **Package** DBeasy

layout_exec.php

This file performs the setting of a layout for a formular.

- **Package** DBeasy

include ["classes.php"](#)*[line 6]*

layout_report_exec.php

Adjust the layout for a report

- **Package** DBeasy

include ["classes.php"](#)*[line 7]*

layout_report_specify.php

The layout for a report can be selected

- **Package** DBeasy

```
function checkLayout($value) [line 23]  
include "functions\_help.php" [line 20]
```

```
include "classes.php" [line 6]
```

layout_specify.php

This file provides the possibility to set and change the layout of a formular.

- **Package** DBeasy

include ["functions_help.php"](#)*[line 21]*

include ["classes.php"](#)*[line 6]*

login.php

This file provides the login interface.

- **Package** DBeasy

login_exec.php

This file performs the login.

- **Package** DBeasy

include "config_user.php" [*line 30*]

include ["SQLFunctions.php"](#)[*line 19*]

include "config_adodb_lib.php" [*line 31*]

include_once ["security_functions.php"](#)[*line 20*]

logo_exec.php

This file performs the setting of a logo for a formular.

- **Package** DBeasy

include ["classes.php"](#)*[line 6]*

include_once ["security_functions.php"](#)*[line 7]*

logo_specify.php

This file provides a HTML input field for the logo path.

- **Package** DBeasy

include ["functions_help.php"](#)*[line 17]*

include ["classes.php"](#)*[line 6]*

mainmenu_formularadmin.php

This file contains the mainmenu on Formadmin level.

- **Package** DBeasy

include ["menu_formularadmin_formular_suche.php"](#)^[line 90]

include ["formular_load.php"](#)^[line 88]

include ["formular_create_exec.php"](#)^[line 86]

include ["formular_create_exec.php"](#)^[line 96]

include ["formular_load.php"](#)^[line 99]

include ["menu_formularadmin_project.php"](#)^[line 105]

include ["menu_formularadmin_formular_bericht.php"](#)^[line 102]

include ["menu_formularadmin_formular_aenderung.php"](#)^[line 80]

include ["formular_load.php"](#)^[line 78]

include ["menu_formularadmin_formular.php"](#)^[line 51]

include ["menu_formularadmin_project.php"](#)^[line 41]

include ["formular_create_exec.php"](#)^[line 65]

include ["formular_load.php"](#)^[line 68]

include ["formular_create_exec.php"](#)^[line 76]

include ["menu_formularadmin_formular_eintrag.php"](#)^[line 71]

include ["security_check_logged_in.php"](#) [line 7]

mainmenu_sysadmin.php

This file contains the mainmenu on Sysadmin level.

- **Package** DBeasy

include ["security_check_logged_in.php"](#)*[line 7]*

menu_formularadmin_formular.php

This file contains the menu for formular administration.

- **Package** DBeasy

include ["security_check_logged_in.php"](#)*[line 6]*

include_once ["SQLFunctions.php"](#)*[line 7]*

menu_formularadmin_formular_aenderung.php

This file contains the submenu for editing a change formular.

- **Package** DBeasy

include ["security_check_logged_in.php"](#)*[line 6]*

menu_formularadmin_formular_bericht.php

This file contains the submenu for editing a report.

- **Package** DBeasy

include ["security_check_logged_in.php"](#)*[line 6]*

menu_formularadmin_formular_eintrag.php

This file contains the submenu for editing a insert formular.

- **Package** DBeasy

include ["security_check_logged_in.php"](#)*[line 6]*

menu_formularadmin_formular_suche.php

This file contains the submenu for editing a search formular.

- **Package** DBeasy

include ["security_check_logged_in.php"](#)*[line 6]*

menu_formularadmin_project.php

This file contains the menu for project administration.

- **Package** DBeasy

include ["security_check_logged_in.php"](#)*[line 6]*

menu_sysadmin_manage_formularadmin.php

This file contains the menu for Formadmin account administration.

- **Package** DBeasy

include ["security_check_logged_in.php"](#)*[line 7]*

menu_sysadmin_manage_sysadmin.php

This file contains the menu for Sysadmin account administration.

- **Package** DBeasy

include ["security_check_logged_in.php"](#)*[line 7]*

preview_aenderung.php

This file generates and displays the preview of a change formular.

- **Package** DBeasy

include **\$complete_path_php** [*line 487*]

include_once ["security_functions.php"](#) [*line 11*]

include_once ["publish_css.php"](#) [*line 10*]

include_once ["filefunctions.php"](#) [*line 9*]

include_once ["FileGenerator.php"](#) [*line 7*]

include_once **"config_adodb_lib.php"** [*line 8*]

include_once ["classes.php"](#) [*line 6*]

preview_eintrag.php

This file generates and displays the preview of a insert formular.

- **Package** DBeasy

include **\$complete_path_php** [line 372]

include_once ["publish_css.php"](#)[line 12]

include_once ["security_functions.php"](#)[line 11]

include_once ["filefunctions.php"](#)[line 10]

include_once ["FileGenerator.php"](#)[line 8]

include_once **"config_adodb_lib.php"** [line 9]

include_once ["classes.php"](#)[line 7]

preview_report.php

This file generates and displays the preview of a report.

- **Package** DBeasy

include_once ["security_functions.php"](#) [line 10]

include_once ["publish_css.php"](#) [line 9]

include_once ["report_functions.php"](#) [line 8]

include_once ["classes.php"](#) [line 6]

include_once ["publish_css.php"](#) [line 7]

function reportIsEmpty() [line 41]

preview_suche.php

This file generates and displays the preview of a search formular.

- **Package** DBeasy

include **\$complete_path_php** [*line 471*]

include_once ["security_functions.php"](#) [*line 12*]

include_once ["publish_css.php"](#) [*line 11*]

include_once ["filefunctions.php"](#) [*line 10*]

include_once ["FileGenerator.php"](#) [*line 8*]

include_once **"config_adodb_lib.php"** [*line 9*]

include_once ["classes.php"](#) [*line 6*]

project_create_exec.php

This file performs the creation of a project.

- **Package** DBeasy

include_once ["FileGenerator.php"](#)*[line 47]*

include_once ["SQLFunctions.php"](#)*[line 46]*

include_once ["security_functions.php"](#)*[line 19]*

project_create_specify.php

This file provides a HTML formular for creating a project.

- **Package** DBeasy

include ["functions_help.php"](#)*[line 19]*

include_once ["SQLFunctions.php"](#)*[line 23]*

project_delete.php

This file performs the delete of a project.

- **Package** DBeasy

include_once ["SQLFunctions.php"](#)*[line 21]*

project_delete_form.php

This file lets you delete a whols Form

- **Package** DBeasy

include_once ["classes.php"](#)*[line 6]*

project_delete_warn.php

This file warns of the delete of a project.

- **Package** DBeasy

project_edit.php

This file offers the possibility to edit and delete a project.

- **Package** DBeasy

include ["functions_help.php"](#)*[line 19]*

include_once ["SQLFunctions.php"](#)*[line 7]*

project_show_formular.php

This file performs the copying of a formular.

There are checks on existence and forbidden names of formulars.

- **Package** DBeasy

include ["functions_help.php"](#)*[line 20]*

include_once ["SQLFunctions.php"](#)*[line 9]*

publish_aenderung.php

This file introduces the publishing of a change formular.

- **Package** DBeasy

include ["classes.php"](#)*[line 6]*

publish_aenderung_exec.php

This file performs the publishing of a change formular.

- **Package** DBeasy

include_once ["publish_css.php"](#)*[line 13]*

include_once ["filefunctions.php"](#)*[line 12]*

include_once ["config_adodb_lib.php"](#) *[line 11]*

include_once ["FileGenerator.php"](#)*[line 10]*

include_once ["classes.php"](#)*[line 6]*

publish_css.php

This file handles all CSS associated functions

- **Package** DBeasy

```
function generateCSSFilename($is_form) [line 264]
function generateForms($style_id, $directory, $which) [line 115]
function generatePreviewForms($style_id, $directory, $which) [line 129]
function generateStyles($style_id, $directory, $which) [line 146]
function getFormStyle($style_id, $which) [line 30]
function getSectionStyle($style_id, $which) [line 76]
function getStyleName($style_id) [line 9]
include_once 'filefunctions.php' [line 6]
```

```
include_once "SQLFunctions.php" [line 7]
```

```
function publishCSS($style_id, $directory) [line 277]
return value: array containing name of CSS files
```

```
function publishFormCSS($style_id, $directory, $filename) [line 163]
function publishReportCSS($style_id, $directory, $filename) [line 209]
```

publish_eintrag.php

This file introduces the publishing of a insert formular.

Remark: The option whether the primary key can be changed by the enduser is set here.

- **Package** DBeasy

include ["security_functions.php"](#)*[line 8]*

include ["classes.php"](#)*[line 7]*

publish_eintrag_exec.php

This file performs the publishing of a insert formular.

- **Package** DBeasy

include_once ["publish_css.php"](#)*[line 11]*

include_once ["filefunctions.php"](#)*[line 9]*

include_once **"config_adodb_lib.php"** *[line 8]*

include_once ["FileGenerator.php"](#)*[line 7]*

include_once ["classes.php"](#)*[line 6]*

publish_report.php

This file introduces the publishing of a report.

- **Package** DBeasy

include_once ["functions_help.php"](#)*[line 32]*

include_once ["security_functions.php"](#)*[line 8]*

include_once ["publish_css.php"](#)*[line 7]*

include_once ["classes.php"](#)*[line 6]*

publish_report_exec.php

This file performs the publishing of a report.

- **Package** DBeasy

include_once ["report_functions.php"](#)*[line 9]*

include_once ["security_functions.php"](#)*[line 8]*

include_once ["publish_css.php"](#)*[line 7]*

include_once ["classes.php"](#)*[line 6]*

publish_search.php

This file introduces the publishing of a search formular.

- **Package** DBeasy

include ["classes.php"](#)*[line 6]*

publish_search_exec.php

This file performs the creation of the search intelligence of the final forms

- **Package** DBeasy

include_once ["publish_css.php"](#)*[line 13]*

include_once ["functions_advanced.php"](#)*[line 14]*

include_once ["filefunctions.php"](#)*[line 12]*

include_once **"config_adodb_lib.php"** *[line 11]*

include_once ["security_functions.php"](#)*[line 9]*

include_once ["FileGenerator.php"](#)*[line 10]*

include_once ["classes.php"](#)*[line 6]*

reportfield_exec.php

The fields to be displayed later on within the report are written into the database

- **Package** DBeasy

include ["classes.php"](#)*[line 7]*

include ["functions_advanced.php"](#)*[line 8]*

include_once ["SQLFunctions.php"](#)*[line 15]*

report_exec.php

Previously specified conditions are actually inserted in this file or existing conditions can be deleted.

- **Package** DBeasy

include_once ["SQLFunctions.php"](#)*[line 52]*

include_once ["SQLFunctions.php"](#)*[line 19]*

include_once ["classes.php"](#)*[line 7]*

report_fieldspecify.php

The fields that should be displayed in a report can be selected here

- **Package** DBeasy

include ["functions_advanced.php"](#)[line 7]

include_once ["functions_help.php"](#)[line 10]

report_functions.php

A collection of functions that are used for publishing and displaying a report

- **Package** DBeasy

function allfields(\$array, \$name) *[line 75]*

Function Parameters:

- *array* **\$array** array The array to be stored
- *string* **\$name** name The name of the new array

Returns a string that can be used to write an the array with its values into a phpfile for later usage

function makeHTML(\$connection, \$query, \$all_fields, \$needed_fields, \$layout, [\$change_formular = ""]) *[line 118]*

Function Parameters:

- *ADODBConnection* **\$connection** connection The adodb connection to be used
- *string* **\$query** query The value to be searched
- *array* **\$all_fields** all_fields All fields in the tables we select
- *array* **\$needed_fields** needed_fields Array with the fields we need
- *string* **\$layout** layout Layout of this form
- *string* **\$change_formular** change_formular Name of the change form attached to this report

Prepares the HTML for a given query

function publishForm(\$form, \$is_preview) *[line 232]*

Function Parameters:

- *int* **\$form** form ID of the form to be published or displayed

- *bool* **\$is_preview** is_preview If it'S a preview or real publishing

Publishes the actual form or displays a preview

function searchKeyInArray(\$key, \$searched_array) [*line 98*]

Function Parameters:

- *array* **\$key** searched_array The array to be searched
- *string* **\$searched_array** key The value to be searched

Returns if a value appears in an array

function write1DimStringArray(\$array, \$name) [*line 13*]

Function Parameters:

- *array* **\$array** array The array to be stored
- *string* **\$name** name The name of the new array

Returns a string that can be used to write an the array with its values into a phfile for later usage

function write2DimStringArray(\$array, \$name) [*line 41*]

Function Parameters:

- *array* **\$array** array The array to be stored
- *string* **\$name** name The name of the new array

Returns a string that can be used to write an 2 dimensional array with its values into a phfile for later usage

report_specify.php

Here addition conditions for a report can be specified

- **Package** DBeasy

include ["functions_advanced.php"](#)[line 7]

include_once ["functions_help.php"](#)[line 10]

search_exec.php

Collects all the necessary Data and then assembles the query

- **Package** DBeasy

include_once ["publish_css.php"](#)*[line 7]*

include_once ["classes.php"](#)*[line 6]*

section_management_exec.php

Executes the rearranging, deleting and editing of sections (Formfield or textfields)

- **Package** DBeasy

function arrayToString2(\$values) [*line 20*]
include ['classes.php'](#) [*line 7*]

section_management_specify.php

This file allows rearranging, deleting and editing of sections (Formfield or textfields)

- **Package** DBeasy

include ["classes.php"](#) [line 7]

include ["functions_help.php"](#) [line 42]

function table_tr(\$values, \$no_array, \$pos) [line 75]

security_check_logged_in.php

This file checks for unauthorized access - meaning without previous, successful login - to a file.

- **Package** DBeasy

security_functions.php

This file contains several functions concerning security of user actions.

Functions control for example whether the user action is forbidden.

- **Package** DBeasy

```
function checkForm($formid) [line 144]  
function compareFields($tables, $fields) [line 71]  
function compareTables($tables) [line 38]  
function fieldError($fields) [line 17]  
function getTablesAndFields($formid) [line 112]  
include_once "filefunctions.php" [line 9]
```

```
include_once "classes.php" [line 8]
```

```
function push_error($error) [line 11]  
function tableError($tables, $fields) [line 25]
```

SQLFunctions.php

This file contains functions concerning database connections and actions.

This file represents DBeasy's interface to ADOdb.

- **Package** DBeasy

```
ADODB_FORCE_NULLS = 1 [line 15]  
include_once "functions\_advanced.php" [line 12]
```

```
include_once "config_adodb_lib.php" [line 10]
```

SQLInstallFunctions.php

SQLInstallFunctions contains special functions for DBeasy installation concerning database issues.

- **Package DBeasy**

include_once ["filefunctions.php"](#)*[line 8]*

include_once ["SQLFunctions.php"](#)*[line 7]*

start.php

This is one of DBeasy's central files. Start.php manages most of the session variables and includes the right files into the frames depending on the context.

- **Package** DBeasy

```
include "security check logged in.php"[line 8]
```

```
include_once "SQLFunctions.php"[line 10]
```

style_editor.php

Here all values for styles can be inserted. Those values are passed on to `style_editor_update` or `style_editor_process` depending if a style is defined for the form or not.

You can only edit a style if you select it for the current form - so we needn't check anything else.

- **Package** DBeasy

```
function arrayAdjustLength($arr, $length) [line 38]  
function ifEqual($var1, $var2, $output) [line 26]  
include "publish\_css.php" [line 24]
```

```
include 'classes.php' [line 9]
```

style_editor_process.php

This page just stores data in the db

- **Package** DBeasy

include ['classes.php'](#) *[line 7]*

function makeInsertArray(\$arr) *[line 22]*

style_editor_update.php
just updates the database with the new values

- **Package** DBeasy

style_index.php

Style index contains the general management interface for styles.

You can select a style for the current form, edit it, delete an existing style or create a new one.

- **Package** DBeasy

```
function getStyleNames() [line 31]
include "functions\_help.php" [line 27]
```

```
include 'classes.php' [line 9]
```

style_index_process_1.php

If you selected a new style in style index it is stored inside the form here.

- **Package** DBeasy

include ["classes.php"](#)*[line 7]*

include_once ["general_functions.php"](#)*[line 21]*

style_index_process_2.php

- **Package** DBeasy

include ["classes.php"](#)*[line 6]*

style_index_process_3.php

This file deletes a style from the database

It has to delete rows in all 3 style tables

- **Package** DBeasy

include ["classes.php"](#)*[line 8]*

table_add_field_exec.php

This file performs the insert of a new table field into a table.

- **Package** DBeasy

include_once ["SQLFunctions.php"](#)*[line 28]*

include_once ["security_functions.php"](#)*[line 20]*

table_add_field_specify.php

This file provides a HTML formular for inserting a new table field into a table.

- **Package** DBeasy

table_add_field_warn.php

This file warns of the insert a new table field into a table.

Reason: Lost of data.

- **Package** DBeasy

include_once "[security functions.php](#) [line 22]

table_create.php

This file provides a HTML formular for creating a table in a database.

Here general information like name of table and number of fields are inserted.

- **Package** DBeasy

```
include "functions\_help.php"[line 20]
```

table_create_exec.php

This file performs the creation of a table in a database.

- **Package** DBeasy

include "config_adodb_lib.php" [*line 7*]

include ["SQLFunctions.php"](#) [*line 9*]

include_once ["security_functions.php"](#) [*line 11*]

table_create_specify.php

This file provides a HTML formular for creating a table in a database.
Here information like field name and type are inserted.

- Package DBeasy

include_once "[security_functions.php](#) [line 24]

table_drop.php

This file performs the delete of a table.

- **Package** DBeasy

include ["SQLFunctions.php"](#)*[line 8]*

table_drop_column.php

This file performs the delete of a table field.

- **Package** DBeasy

include_once ["SQLFunctions.php\[line 8\]](#)

table_drop_column_warn.php

This file warns of the delete of a table field.

- **Package** DBeasy

table_drop_warn.php

This file warns of the delete of a table.

- **Package** DBeasy

table_show_structure.php

This file provides a HTML formular for editing and deleting a table fields.

- **Package** DBeasy

include ["functions_help.php"](#)*[line 22]*

include ["table_add_field_specify.php"](#)*[line 85]*

include_once ["SQLFunctions.php"](#)*[line 26]*

template.php

This file is a template file for all kind of user interface files .

- **Package** DBeasy

include ["functions_help.php"](#)*[line 19]*

include ["classes.php"](#)*[line 7]*

uninstall.php

This file warns of the deinstallation of DBeasy.

- **Package** DBeasy

uninstall_exec.php

This file performs the deinstallation of DBeasy.

- **Package** DBeasy

include ["SQLFunctions.php"](#)*[line 16]*

user_db_mapping.php

This file provides a HTML formular for association of a database with an user.

- **Package** DBeasy

include ["functions_help.php"](#)*[line 24]*

include_once ["SQLFunctions.php"](#)*[line 8]*

user_db_mapping_connection.php

This file includes - depending of the database type - a special HTML formular where the access data can be inserted.

- **Package** DBeasy

include ["connection/standard_connection_form.php"](#) [line 48]

include ["connection/standard_connection_form.php"](#) [line 50]

include ["connection/access_connection_form.php"](#) [line 46]

include ["connection/sqlite_connection_form.php"](#) [line 44]

include ["connection/odbc_connection_form.php"](#) [line 42]

include ["functions_help.php"](#) [line 22]

user_db_mapping_delete.php

This file performs the delete of an association of a database with an user.

- **Package** DBeasy

include_once ["SQLFunctions.php\[line 7\]](#)

user_db_mapping_exec.php

This file performs the association of a database with an user.

- **Package** DBeasy

include_once ["security_functions.php"](#)*[line 8]*

include_once ["SQLFunctions.php"](#)*[line 7]*

user_db_mapping_specify.php

This file provides a HTML formular for choosing the database type during the step of association of a database with an user.

- **Package** DBeasy

```
include "functions\_help.php"[line 31]
```

```
include_once "SQLFunctions.php"[line 7]
```

user_db_mapping_warn.php

This file warns the user of the delete of an association of a database with an user.

- **Package** DBeasy

user_delete.php

This file performs the delete of an user account.

- **Package** DBeasy

include_once ["SQLFunctions.php\[line 6\]](#)

user_delete_warn.php

This file warns the user of the delete of an user account.

- **Package** DBeasy

include_once ["SQLFunctions.php"](#)*[line 17]*

user_dir_mapping.php

This file provides a HTML formular for association of a directory with an user.

- **Package** DBeasy

include ["SQLFunctions.php"](#)*[line 17]*

user_dir_mapping_delete.php

This file performs the delete of an association of a directory with an user.

- **Package** DBeasy

include_once ["SQLFunctions.php\[line 7\]](#)

user_dir_mapping_exec.php

This file performs the association of a directory with an user.

- **Package** DBeasy

include ["SQLFunctions.php"](#)*[line 7]*

include_once ["security_functions.php"](#)*[line 8]*

user_dir_mapping_warn.php

This file warns the user of the delete of an association of a directory with an user.

- **Package** DBeasy

user_entry_exec.php

This file performs the creation of an user account.

- **Package** DBeasy

include_once ["security_functions.php"](#)*[line 22]*

include_once ["SQLFunctions.php"](#)*[line 21]*

user_entry_specify.php

This file provides a HTML formular for creating an user account.

- **Package** DBeasy

include ["functions_help.php"](#)*[line 21]*

user_show_table.php

This file provides a HTML formular for editing and deleting an user account.

- **Package** DBeasy

include ["functions_help.php"](#)*[line 22]*

include_once ["SQLFunctions.php"](#)*[line 9]*

Package DBeasy Classes

Class Error

[line 126]

Output and retrieve error messages

Currently they are stored in a file - might be changed later

- **Package** DBeasy

Error::\$filename

mixed = [line 128]

Constructor function Error::Error() *[line 133]*

Constructor

function Error::addError(\$error_message) *[line 152]*

Adds an error for later output

function Error::deleteFile() *[line 141]*

Deletes the error file

function Error::getErrors() *[line 162]*

returns the errors

Class FileGenerator

[line 16]

FileGenerator class contains functions to create, close and writing into a file

- **Package** DBeasy

FileGenerator::\$file_pointer

mixed = [line 19]

function FileGenerator::closeFile() *[line 37]*

Closes a file

function FileGenerator::createFile(\$name_of_file) *[line 29]*

Function Parameters:

- *string* **\$name_of_file** Name of file being created

Creates and begins a file

list function FileGenerator::my_split(\$string, \$pattern) *[line 57]*

Function Parameters:

- *string* **\$string** String to be splitted
- *string* **\$pattern** Specific pattern where given string is splitted into tokens

Splits a string into tokens at some specific points

function FileGenerator::write(\$text) *[line 46]*

Function Parameters:

- *string* **\$text** Text being written into file

Writes text into the file

Class fileinput

[line 12]

Loads data from textfiles

- **Package** DBeasy

fileinput::\$files

mixed = [line 15]

Constructor function fileinput::fileinput(\$filename) *[line 21]*

Function Parameters:

- *string* **\$filename** filename Filename

Opens a file

function fileinput::close() *[line 51]*

Closes the file

function fileinput::read() *[line 29]*

Reads a single line

function fileinput::readAll() *[line 38]*

Reads an entire file into an array.

Each line in the document is a field in the array.

Class fileoutput

[line 62]

Output data to textfiles

- **Package** DBeasly

fileoutput::\$files

mixed = [line 65]

Constructor function fileoutput::fileoutput(\$filename, \$should_overwrite) [line 72]

Function Parameters:

- *string* **\$filename** filename Filename
- *bool* **\$should_overwrite** should_overwrite an existent file may be overwritten

Opens a file

function fileoutput::close() [line 114]

Closes the file

function fileoutput::write(\$string) [line 90]

Function Parameters:

- *string* **\$string** string the data we want to output

Write data to the file

function fileoutput::writeln(\$string) [line 102]

Function Parameters:

- *string* **\$string** string the data we want to output

Writes a line to the file

Class Formular

[line 344]

Contains all data of a formular

- **Package** DBeasy

Formular::\$debug

mixed = [line 357]

Formular::\$dependency

mixed = [line 353]

Formular::\$headline

mixed = [line 355]

Formular::\$id

mixed = [line 346]

Formular::\$layout

mixed = [line 350]

Formular::\$logic

mixed = [line 352]

Formular::\$logo

mixed = [line 354]

Formular::\$name

mixed = [line 348]

Formular::\$project_id

mixed = [line 347]

Formular::\$sections

mixed = [line 356]

Formular::\$style

mixed = [line 351]

Formular::\$type

mixed = [line 349]

function Formular::addEnumeration(\$pos, \$enumerations) *[line 811]*

Function Parameters:

- *int* **\$pos** pos Position of the section
- *stringarray* **\$enumerations** enumerations Enumeration we want to add

Adds a Enumertion to a section

function Formular::addFormField(\$sec_data_type, \$sec_label, \$sec_appearance, \$sec_table, \$sec_field) *[line 571]*

Function Parameters:

- *char* **\$sec_data_type** sec_data_type datatype of the formfield in ADODB notation (single char)
- *string* **\$sec_label** sec_label Label of the formfield
- *string* **\$sec_appearance** sec_appearance Appearance of the Form field (textbox, textarea, radiobutton, dropdownmenu, checkbox)
- *string* **\$sec_table** sec_table Name of the table
- *string* **\$sec_field** sec_field Name of the field

Adds a formfield by creating a section

function Formular::addTextField(\$sec_text) *[line 501]*

Function Parameters:

- *string* **\$sec_text** sec_text Text of the section

Adds a textfield by creating a section and setting the textfield values

function Formular::create(\$project_id2, \$form_type, \$form_name) [*line 376*]

Function Parameters:

- *int* **\$project_id2** project_id2 ID of the project where this form should be added to
- *string* **\$form_type** form_type type of the form (bericht[=report],suche[=search],eintrag[=input],aenderung[=change],
- *string* **\$form_name** form_name name of the form

inserts into DB and generates new ID

function Formular::deleteForm() [*line 822*]

Deletes the form

function Formular::deleteSection(\$position) [*line 759*]

Function Parameters:

- *int* **\$position** pos Position of the section to be deleted

Deletes a section

function Formular::existsSection(\$table, \$field) [*line 546*]

Function Parameters:

- *string* **\$table** sec_table Name of the table
- *string* **\$field** sec_field Name of the field

checks if a Section exists for a certain DB field

function Formular::formular() [*line 364*]

Constructor

We just adjust the debug output and nothing else here...

function Formular::getSection(\$sec_id) [*line 720*]

Function Parameters:

- *int* **\$sec_id** sec_id ID of the section

Returns the values of the section

function Formular::getSections() [*line 734*]

Returns the values of all sections of this form

function Formular::getValues() [*line 748*]

Returns the values of this form

function Formular::load(\$form_id) [*line 411*]

Function Parameters:

- *int* **\$form_id** form_id ID of the form that should be loaded

loads the form with the ID from the SystemDB

function Formular::moveSectionDown(\$pos) [*line 628*]

Function Parameters:

- *int* **\$pos** pos Position of the section to be moved

Moves the section down

function Formular::moveSectionUp(\$pos) [*line 594*]

Function Parameters:

- *int* **\$pos** pos Position of the section to be moved

Moves the section up

function Formular::setDependency(\$new_dependency) [*line 684*]

Function Parameters:

- *int* **\$new_dependency** new_dependency new dependency of the formular

Sets a dependency

function Formular::setEnumeration(\$position, \$enumerations) [*line 789*]

Function Parameters:

- *int* **\$position** pos Position of the section
- *stringarray* **\$enumerations** enumerations Enumeration we want to set

Modifies a enumeration by deleting the old one and adding a new one with the new values

function Formular::setFormField(\$position, \$sec_data_type, \$sec_label, \$sec_appearance, \$sec_table, \$sec_field) [*line 536*]

Function Parameters:

- *int* **\$position** position Position of the section to be modified
- *char* **\$sec_data_type** sec_data_type datatype of the formfield in ADODB notation (single char)
- *string* **\$sec_label** sec_label Label of the formfield
- *string* **\$sec_appearance** sec_appearance Appearance of the Form field (textbox, textarea, radiobutton, dropdownmenu, checkbox)
- *string* **\$sec_table** sec_table Name of the table
- *string* **\$sec_field** sec_field Name of the field

Modifies a formfields values, at postion \$position

function Formular::setHeadline(\$new_headline) [*line 711*]

Function Parameters:

- *int* **\$new_headline** new_headline new headline of the formular

Sets a headline

function Formular::setLayout(\$new_layout) [*line 666*]

Function Parameters:

- *string* **\$new_layout** new_layout new layout of the formular

Sets a layout

function Formular::setLogo(\$new_logo) [*line 702*]

Function Parameters:

- *int* **\$new_logo** new_logo new logo of the formular

Sets a logo

function Formular::setName(\$new_name) [*line 693*]

Function Parameters:

- *int* **\$new_name** new_name new name of the formular

Sets a name

function Formular::setStyle(\$new_style) [*line 675*]

Function Parameters:

- *string* **\$new_style** new_style new Style of the formular

Sets a style

function Formular::setTextField(\$position, \$sec_text) *[line 521]*

Function Parameters:

- *int* **\$position** position Position of the textfield
- *sec_text* **\$sec_text** New text

Modifies the textfield's text at postion \$position

function Formular::updateDB() *[line 458]*

updates the database with the current values of the variables

Class HTMLGenerator

[line 74]

HTMLGenerator class contains functions to create a HTML file.

- **Package** DBeasy

function HTMLGenerator::makeBR(\$number) *[line 115]*

Function Parameters:

- *int* **\$number** Number of repetition

Creates a row break

function HTMLGenerator::makeChangePrimaryKeyLink(\$filename) *[line 476]*

Function Parameters:

- *string* **\$filename** Name of file where primary key can be changed

Offers a link to change the primary key. If the primary key was changed, the value is displayed.

function HTMLGenerator::makeCheckbox(\$name, \$value, [\$remember_value = false]) *[line 321]*

Function Parameters:

- *string* **\$name** Name of checkbox
- *string* **\$value** Value of checkbox
- *bool* **\$remember_value** Flag indicating whether checkbox is pre-given with old value or not

Creates a checkbox

function HTMLGenerator::makeDropDownMenu(\$select_name, \$option_array) *[line 359]*

Function Parameters:

- *string* **\$select_name** Name of dropdown menu
- *option_array* **\$option_array** Array having following structure: value => description

Creates a DropDown menu

function HTMLGenerator::makeEasyDropDownMenu(\$select_name, \$option_array) *[line 378]*

Function Parameters:

- *string* **\$select_name** Name of dropdown menu
- *option_array* **\$option_array** 1-dim-array containing the options

Creates a DropDown menu where option value and description are the same

function HTMLGenerator::makeFormBegin(\$form_name, \$form_action) *[line 208]*

Function Parameters:

- *string* **\$form_name** Name of formular

- *string* **\$form_action** Executing file

Creates beginning of a HTML formular (POST method)

function HTMLGenerator::makeFormEnd() *[line 230]*

Finishes a HTML formular

function HTMLGenerator::makeFormGetBegin(\$form_name, \$form_action) *[line 220]*

Function Parameters:

- *string* **\$form_name** Name of formular
- *string* **\$form_action** Executing file

Creates beginning of a HTML formular (GET method)

function HTMLGenerator::makeFormHead(\$logo_link, \$formular_title) *[line 164]*

Function Parameters:

- *string* **\$logo_link** Link to the location of logo
- *string* **\$formular_title** Title of the formular

Creates headline consisting of logo and title

function HTMLGenerator::makeFormularButton(\$type, \$value) *[line 242]*

Function Parameters:

- *string* **\$type** Type of buttons, i.e. "submit" or "reset"
- *string* **\$value** Value of buttons

Creates a HTML formular button

function HTMLGenerator::makeHTMLEnd() *[line 97]*

Finishes HTML document and adds end tags

function HTMLGenerator::makeHTMLHead(\$title, \$style_file_location) *[line 86]*

Function Parameters:

- *string* **\$title** Title of HTML document
- *string* **\$style_file_location** Location of CSS file

Creates HTML header

function HTMLGenerator::makeLabel(\$for, \$label) *[line 341]*

Function Parameters:

- *string* **\$for** Reference; name of HTML formular element belonging to this label
- *string* **\$label** Descriptive label text

Creates a label for a HTML formular element

function HTMLGenerator::makeMultiLineInputField(\$name, \$cols, \$rows, [\$remember_value = false]) *[line 286]*

Function Parameters:

- *string* **\$name** Name of input-field
- *int* **\$cols** Number of columns
- *int* **\$rows** Number of rows
- *bool* **\$remember_value** Flag indicating whether input-field contains old value or not

Creates multi-line input-field

function HTMLGenerator::makeRadioButton(\$name, \$value) *[line 307]*

Function Parameters:

- *string* **\$name** Name of radio button
- *string* **\$value** Value of radio button

Creates a radio button

function HTMLGenerator::makeReturnToIndexLink(\$path) [*line 545*]

Function Parameters:

- *string* **\$path** Path of project index

Creates Link to the project index

function HTMLGenerator::makeSingleLineInputField(\$name, \$size, \$maxlength, [\$remember_value = false]) [*line 264*]

Function Parameters:

- *string* **\$name** Name of input-field
- *int* **\$size** Size of input-field
- *int* **\$maxlength** Max. size of input-field
- *bool* **\$remember_value** Flag indicating whether input-field contains old value or not

Creates single-line input-field

function HTMLGenerator::makeTableBegin() [*line 400*]

Creates opening tag for a table

function HTMLGenerator::makeTableEnd() [*line 410*]

Creates finishing tag for a table

function HTMLGenerator::makeTDBegin() [*line 440*]

Creates opening tag for a table cell

function HTMLGenerator::makeTDColspanBegin(\$no) [*line 461*]

Function Parameters:

- *int* **\$no** Number of table cells forming one table cell

Creates opening tag for a table cell ranging over several table cells

function HTMLGenerator::makeTDEnd() [*line 450*]

Creates finishing tag for a table cell

function HTMLGenerator::makeTextField(\$text) [*line 187*]

Function Parameters:

- *string* **\$text** Text being written between paragraph tags

Creates paragraph

function HTMLGenerator::makeTRBegin() [*line 420*]

Creates opening tag table row

function HTMLGenerator::makeTREnd() [*line 430*]

Creates finishing tag for a table row

function HTMLGenerator::makeULLink(\$value_array) [*line 131*]

Function Parameters:

- *value_array* **\$value_array** Variable contains the elements of the unordered list

Creates a unordered list

function HTMLGenerator::transmitChangedPrimaryKeyValue(\$primary_key_for_change_formular) [*line 488*]

Function Parameters:

- *array* **\$primary_key_for_change_formular** Array containing primary keys in format:
TABLE_TABLEFIELD

Creates a table with result of changed primary keys after the change_id_formular has been used

function HTMLGenerator::writehtml(\$text) [*line 149*]

Function Parameters:

- *string* **\$text** Text being written into document

Writes arbitrary string into the HTML document

Class PHPGenerator

[line 560]

PHPGenerator class contains functions to create a executing PHP file.

- **Package** DBeasy

function PHPGenerator::makeAutoReturn(\$link) *[line 585]*

function PHPGenerator::makeChangeIDFormular(\$primary_keys, \$all_formular_fields, \$filename_of_insert_formular) *[line 906]*

Function Parameters:

- *array* **\$primary_keys** Array containing possible changeable primary keys
- *string* **\$all_formular_fields** Name of insert formular that is loaded

Creates the formular for changing the primary key

function PHPGenerator::makeConnection(\$db_type, \$db_address, \$db_username, \$db_password, \$db_name, \$db_path, \$db_dsn) *[line 639]*

Function Parameters:

- *string* **\$db_type** Type of database
- *string* **\$db_address** Location of database
- *string* **\$db_username** Username for database access
- *string* **\$db_password** Password for database access

- *string* **\$db_name** Name of database
- *string* **\$db_path** Location of database
- *string* **\$db_dsn** DSN of database

Creates connection statements to database

function PHPGenerator::makeFormHead(\$logo_link, \$formular_title) [*line 889*]

Function Parameters:

- *string* **\$logo_link** Link to the location of logo
- *string* **\$formular_title** Title of the formular

Creates headline consisting of logo and title

function PHPGenerator::makeHTMLHead(\$title, \$style_file_location) [*line 876*]

Function Parameters:

- *string* **\$title** Title of HTML document
- *string* **\$style_file_location** Location of CSS file

Creates HTML header

function PHPGenerator::makeIncludeAdo() [*line 622*]

Creates an include of ADOdb files

function PHPGenerator::makeInputCheck(\$returnlink) [*line 608*]

Function Parameters:

- *string* **\$returnlink** Page where to return

Introducing SQL injection test

function PHPGenerator::makePHPBegin() [*line 566*]

Creates beginning of PHP file and adds comments about date and author of this PHP file

function PHPGenerator::makePHPEnd() [*line 577*]

Finishes a PHP file

function PHPGenerator::makeReturnButton(\$link) [*line 597*]

Function Parameters:

- *string* **\$link** Path of file the back link is pointing to

Creates a back link

function PHPGenerator::makeReturnToIndexLink(\$path) [*line 968*]

Function Parameters:

- *string* **\$path** Path of project index

Creates Link to the project index

function PHPGenerator::makeSQL_insert(\$name_array) [*line 670*]

Function Parameters:

- *name_array* **\$name_array** Array containing the table-fields that are used in the Insert formular

Creates the executing PHP file for an Insert formular

function PHPGenerator::makeSQL_update(\$all_table_fields, \$formular_fields_used) [*line 792*]

Function Parameters:

- *array* **\$all_table_fields** Array containing ALL fields of a table on that the Change formular refers to (in form of table__field)
- *array* **\$formular_fields_used** Array containing the table__field used in Change formular

Creates the executing PHP file for a Change formular

Class Section

[line 20]

A section is a part of a formular. Each section belongs to just one formular so it stores its formular ID.

It contains all values for formfields or textfields.

- **Package DBeasy**

Section::\$appearance

mixed = [line 33]

Section::\$data_type

mixed = [line 31]

Section::\$debug

mixed = [line 28]

Section::\$enumerations

mixed = [line 36]

Section::\$field

mixed = [line 35]

Section::\$form_id

mixed = [line 25]

Section::\$id

mixed = [line 24]

Section::\$label

mixed = [line 32]

Section::\$my_sql

mixed = [line 27]

Section::\$position

mixed = [line 26]

Section::\$table

mixed = [line 34]

Section::\$text

mixed = [line 39]

Constructor function Section::Section() *[line 47]*

Section constructor

Constructor - just allows us to enable or disable the debugging output

function Section::addEnumeration(\$enumerations) *[line 274]*

Function Parameters:

- *stringarray* **\$enumerations** Enumarations we want to add

Adds enumertions to this section, creates table if necessary

function Section::createSection(\$sec_form_id, \$sec_position) *[line 58]*

Function Parameters:

- *int* **\$sec_form_id** sec_form_id ID of the form we're working on
- *int* **\$sec_position** sec_position position of the section we're working on

Creates a section for a certain form at a certain position

function Section::deleteEnumeration() *[line 319]*

deletes the enumeration of this section

function Section::deleteSection() *[line 229]*

Deletes a section together with its enumeration

function Section::getValues() [*line 159*]

Returns the values of this sections in an array with:

id,form_id,data_type,label,appearance,position,text,table,field and enumerations

function Section::isText() [*line 149*]

Returns if this is a textfield

function Section::loadEnumerations() [*line 251*]

Loads the enumeration for this section

function Section::loadSection(\$sec_id) [*line 83*]

Function Parameters:

- *int* **\$sec_id** sec_id ID of the section that should be loaded

loads a section with a certain section ID

function Section::setFormField(\$sec_data_type, \$sec_label, \$sec_appearance, \$sec_table, \$sec_field) [*line 122*]

Function Parameters:

- *char* **\$sec_data_type** sec_data_type datatype of the formfield in ADODB notation (single char)
- *string* **\$sec_label** sec_label Label of the formfield
- *string* **\$sec_appearance** sec_appearance Appearance of the Form field (textbox, textarea, radiobutton, dropdownmenu, checkbox)
- *string* **\$sec_table** sec_table Name of the table
- *string* **\$sec_field** sec_field Name of the field

Sets the parameters for a formfield

function Section::setPosition(\$pos) [*line 182*]

Adjusts the position

function Section::setTextField(\$sec_text) [*line 141*]

Sets the parameter for a textfield

function Section::updateDB() [*line 190*]

Updates the DB with the values of this section

Class SQLFunctions

[*line 20*]

Contains all functions which handle with SQL

- **Package DBeasy**

SQLFunctions::\$connection

mixed = [line 23]

SQLFunctions::\$current_result

mixed = "empty" [line 32]

SQLFunctions::\$db_address

mixed = [line 25]

SQLFunctions::\$db_dsn

mixed = [line 30]

SQLFunctions::\$db_name

mixed = [line 28]

SQLFunctions::\$db_password

mixed = [line 27]

SQLFunctions::\$db_path

mixed = [line 29]

SQLFunctions::\$db_type

mixed = [line 24]

SQLFunctions::\$db_username

mixed = [line 26]

SQLFunctions::\$debug

mixed = [line 22]

Constructor function SQLFunctions::SQLFunctions() *[line 34]*

function SQLFunctions::addField(\$table_name, \$fields) *[line 518]*

Function Parameters:

- *string* **\$table_name** Name of table
- *stringarray* **\$fields** Name and type of fields

Adds an field to the table mentioned in the arguments

string function SQLFunctions::addUser(\$login, \$password, \$user_group, \$firstname, \$lastname) *[line 263]*

Function Parameters:

- *string* **\$login** Login name
- *string* **\$password** Password
- *int* **\$user_group** User group : for group sysadmin 1, for group formadmin 2
- *string* **\$firstname** First name
- *string* **\$lastname** Last name

Adds an user to the table dbeasy_user

function SQLFunctions::autoConnectToDB() *[line 120]*

Connects to the Database specified in the file config_user.php

- **See** config_user.php

function SQLFunctions::connectToDB(\$db_type, \$db_address, \$db_username, \$db_password, \$db_name, \$db_path, \$db_dsn) *[line 60]*

Function Parameters:

- *string* **\$db_type** Database-Type
- *string* **\$db_address** Database-Address
- *string* **\$db_username** Username for connect to database
- *string* **\$db_password** Password for connect to database
- *string* **\$db_name** Name of database
- *string* **\$db_path** Path of database
- *string* **\$db_dsn** DSN of database

Connects to the Database specified in the arguments of the function

Not all parameters are necessary, that is dependent on the database type

function SQLFunctions::connectToDBODBM() [*line 191*]

Abbreviation: connectToDataBaseOverDataBaseMapping

This function includes: 1) a connection establishment to the DBEasy's configuration database 2) query for access data (dbeasy_user_db_mapping) for selected user and selected database 3) a connection establishment to the selected database with access data of point 2)

string function SQLFunctions::createTable(\$table_name, \$fields) [*line 211*]

Function Parameters:

- *string* **\$table_name** Name of table
- *stringarray* **\$fields** Name of all fields of table

Creates a table which name and fields are given in the argument of the function

recordSet function SQLFunctions::deleteRow(\$table_name, \$condition) [*line 466*]

Function Parameters:

- *string* **\$table_name** Name of table
- *string* **\$condition** Condition

Deletes the row in a table which satisfies the given condition

If all rows shall be deleted, fill in "" as \$condition

function SQLFunctions::dropColumn(\$table_name, \$column_name) [*line 622*]

Function Parameters:

- *string* **\$table_name** Name of table
- *string* **\$column_name** Name of Column

Deletes the specified Column from the Table

function SQLFunctions::dropTable(\$table_name) [*line 742*]

Function Parameters:

- *string* **\$table_name** Name of table

Deletes a table from the database

resultSet function SQLFunctions::executeSQL(\$sql_statement) [*line 237*]

Function Parameters:

- *string* **\$sql_statement** SQL-Statement

Executes the SQL-Statement given as an argument of the function

function SQLFunctions::existsItem(\$table_name, \$key, \$key_value) [*line 811*]

date function SQLFunctions::generateDate() [*line 341*]

Generates the Date, dependent on the databasetype

function SQLFunctions::generateID(\$seq_name, [\$start_value = "1"], 2) [*line 864*]

Function Parameters:

- *int* **2** Unique ID

- *string* **\$seq_name** Name of sequence
- *int* **\$start_value** Start value, by default set to 1

Generates an unique ID

stringarray function SQLFunctions::getFields(\$table_name) [*line 791*]

Function Parameters:

- *string* **\$table_name** Name of table

Returns all fields of the specified table in an array

string function SQLFunctions::getIntPrimaryKey(\$table, \$array_possible_primary_key) [*line 827*]

Function Parameters:

- *string* **\$table** Name of table
- *stringarray* **\$array_possible_primary_key** Name of possible primary keys

Returns the name of primary key, if there is only one primary key and primary key is of type integer, else false

int function SQLFunctions::getMax(\$table_name, \$field) [*line 491*]

Function Parameters:

- *string* **\$table_name** Name of table
- *string* **\$field** Name of field

Returns the maximal number of datarecords, which are in the table

function SQLFunctions::getResultMsg() [*line 42*]

Returns the Result Message

function SQLFunctions::insertData(\$table_name, \$values_array) [*line 417*]

Function Parameters:

- *string* **\$table_name** Name of table
- *stringarray* **\$values_array** All values which shall be inserted

Inserts an Datarecord into a table. \$value_array is an array which contains the new values.

The values need to have the same order in the array and in the database.

`function SQLFunctions::LogTable($table_name) [line 755]`

Function Parameters:

- *string* **\$table_name** Name of table

Adds the tablename to the dbeasy_log_systables, needed for uninstallation

`recordSet function SQLFunctions::selectAll($table_name, $condition) [line 381]`

Function Parameters:

- *string* **\$table_name** Name of table
- *string* **\$condition** Condition, "" if no condition is specified

Returns all Datarecords of a table, which satisfy the given condition in an array.

If there is no condition, fill in "" as the \$condition argument

`stringarray function SQLFunctions::showTables() [line 505]`

Shows all tables, which are contained in the database to which the programm is connected

`function SQLFunctions::tableExists($table_name, 1) [line 770]`

Function Parameters:

- *boolean* **1** True or False
- *string* **\$table_name** Name of table

Checks if the specified table exists in the database

function SQLFunctions::updateRow(\$table_name, \$key, \$key_value, \$fields_array) [*line 291*]

Function Parameters:

- *string* **\$table_name** Name of table
- *string* **\$key** Key
- *string* **\$key_value** Key value
- *stringarray* **\$fields_array** Name of fields

Updates an existing row, with the values in the variable \$fields_array

Class SQLInstallFunctions

[*line 13*]

SQLInstallFunctions contains special functions for DBeasy installation concerning database issues.

- **Package** DBeasy

Result-Message function SQLInstallFunctions::createConditionTable(\$connection) [*line 145*]

Function Parameters:

- *connection* **\$connection** Connection pointer

Creates DBeasy sytem table: Condition

Result-Message function `SQLInstallFunctions::createFormularTable($connection)` [*line 76*]

Function Parameters:

- *connection* **\$connection** Connection pointer

Creates DBeasy sytem table: Formular

Result-Message function `SQLInstallFunctions::createHelpTable($connection)` [*line 262*]

Function Parameters:

- *connection* **\$connection** Connection pointer

Creates DBeasy sytem table: Help

Result-Message function `SQLInstallFunctions::createJoinTable($connection)` [*line 223*]

Function Parameters:

- *connection* **\$connection** Connection pointer

Creates DBeasy sytem table: Join

Result-Message function `SQLInstallFunctions::createLogicTable($condition)` [*line 158*]

Function Parameters:

- *connection* **\$condition** Connection pointer

Creates DBeasy sytem table: Logic

Result-Message function `SQLInstallFunctions::createLogTable($connection)` [*line 131*]

Function Parameters:

- *connection* **\$connection** Connection pointer

Creates DBeasy sytem table: Logging

Result-Message function SQLInstallFunctions::createProjectTable(\$connection) [*line 63*]

Function Parameters:

- *connection* **\$connection** Connection pointer

Creates DBeasy sytem table: Project

Result-Message function SQLInstallFunctions::createReportCondTable(\$connection) [*line 236*]

Function Parameters:

- *connection* **\$connection** Connection pointer

Creates DBeasy sytem table: ReportCondition

Result-Message function SQLInstallFunctions::createReportFieldTable(\$connection) [*line 249*]

Function Parameters:

- *connection* **\$connection** Connection pointer

Creates DBeasy sytem table: ReportField

Result-Message function SQLInstallFunctions::createSectionTable(\$connection) [*line 92*]

Function Parameters:

- *connection* **\$connection** Connection pointer

Creates DBeasy sytem table: Section

Result-Message function SQLInstallFunctions::createStyleFormsTable(\$connection) [*line 184*]

Function Parameters:

- *connection* **\$connection** Connection pointer

Creates DBeasy sytem table: StyleForm

Result-Message function SQLInstallFunctions::createStyleSectionsTable(\$connection) [*line 197*]

Function Parameters:

- *connection* **\$connection** Connection pointer

Creates DBeasy sytem table: StyleSection

Result-Message function SQLInstallFunctions::createStyleStylesTable(\$connection) [*line 210*]

Function Parameters:

- *connection* **\$connection** Connection pointer

Creates DBeasy sytem table: StyleStyle

Result function SQLInstallFunctions::createUserDBDataFile(\$db_type, \$db_address, \$db_username, \$db_password, \$db_name, \$db_path, \$db_dsn) [*line 28*]

Function Parameters:

- *string* **\$db_type** Type of database
- *string* **\$db_address** Address of database
- *string* **\$db_username** Username for database access
- *string* **\$db_password** Password for database access
- *string* **\$db_name** Name of database
- *string* **\$db_path** Path of database
- *string* **\$db_dsn** DSN of database

Creates a php file containing database access data for DBeasy's system tables

Result-Message function SQLInstallFunctions::createUserDBMappingTable(\$connection) [*line 105*]

Function Parameters:

- *connection* **\$connection** Connection pointer

Creates DBeasy sytem table: UserDBMapping

Result-Message function SQLInstallFunctions::createUserDirMappingTable(\$connection) [*line 118*]

Function Parameters:

- *connection* **\$connection** Connection pointer

Creates DBeasy sytem table: UserDirMapping

Result-Message function SQLInstallFunctions::createUserTable(\$connection) [*line 50*]

Function Parameters:

- *connection* **\$connection** Connection pointer

Creates DBeasy sytem table: User

Result-Message function SQLInstallFunctions::createWorkfieldTable(\$condition) [*line 171*]

Function Parameters:

- *connection* **\$condition** Connection pointer

Creates DBeasy sytem table: Workfield

Appendices

Appendix A - Class Trees

Package DBeasy

Error

- [Error](#)

FileGenerator

- [FileGenerator](#)
 - [HTMLGenerator](#)
 - [PHPGenerator](#)

fileinput

- [fileinput](#)

fileoutput

- [fileoutput](#)

Formular

- [Formular](#)

Section

- [Section](#)

SQLFunctions

- [SQLFunctions](#)
 - [SQLInstallFunctions](#)

Index

A

ADODB_FORCE_NULLS	102
arrayAdjustLength()	105
arrayToString2()	98
allfields()	94
<i>Returns a string that can be used to write an the array with its values into a phpfile for later usage</i>	
arrayToStrings()	27
<i>Converts an array into a string with ; as a seperator</i>	
arrayToString()	35
<i>Returns a string built from an array</i>	
access_connection_form.php	6
<i>This file displays input fields required to connect to access</i>	

C

compareTables()	101
compareFields()	101
checkForm()	101
checkLayout()	54
constructor Error::Error()	141
<i>Constructor</i>	
constructor fileinput::fileinput()	143
<i>Opens a file</i>	
constructor SQLFunctions::SQLFunctions()	164
constructor Section::Section()	161
<i>Section constructor</i>	
constructor fileoutput::fileoutput()	144
<i>Opens a file</i>	
copyReportToForm()	33
<i>Copies reports into reports</i>	
copyFormToReport()	32
<i>Copies non reports into reports</i>	
connection_error.php	10
<i>This file appears if there is a problem with connecting to the database</i>	
classes.php	5
<i>Core classes (Section and Formular are defined here)</i>	
choose_tab_field_exec.php	4
<i>Here the actual formfield section is created and added to the database</i>	
create_textfield.php	11
<i>This file provides the possibility to create a textfield in a formular.</i>	
create_textfield_process.php	12
<i>This file performs the creation of a textfield in a formular.</i>	
copyFormToForm()	32

<i>Copies non reports into non reports</i>	
checkNameExists()	32
clearDependency()	27
<i>Clears the dependency of forms that have a given type and dependency</i>	
choose_tab_field.php	3
<i>Selecting an formfield with all necessary values:</i>	

D

dbeasy_security.php	14
<i>This file contains some functions for SQL injection security.</i>	
database_show_table.php	13
<i>This file provides an overview of the tables within a selected database.</i>	

E

Error::getErrors()	141
<i>returns the errors</i>	
Error::deleteFile()	141
<i>Deletes the error file</i>	
Error::addError()	141
<i>Adds an error for later output</i>	
Error::\$filename	141
Error	141
<i>Output and retrieve error messages</i>	

F

Formular::addTextField()	146
<i>Adds a textfield by creating a section and setting the textfield values</i>	
Formular::addFormField()	146
<i>Adds a formfield by creating a section</i>	
Formular::addEnumeration()	146
<i>Adds a Enumertion to a section</i>	
Formular::create()	147
<i>inserts into DB and generates new ID</i>	
Formular::deleteForm()	147
<i>Deletes the form</i>	
Formular::existsSection()	147
<i>checks if a Section exists for a certain DB field</i>	
Formular::deleteSection()	147
<i>Deletes a section</i>	
Formular::\$type	146
Formular::\$style	146
Formular::\$logic	145
Formular::\$layout	145
Formular::\$id	145
Formular::\$logo	145
Formular::\$name	146
Formular::\$sections	146

Formular::\$project_id	146
Formular::formular()	148
<i>Constructor</i>	
Formular::getSection()	148
<i>Returns the values of the section</i>	
Formular::setLogo()	150
<i>Sets a logo</i>	
Formular::setLayout()	150
<i>Sets a layout</i>	
Formular::setHeadline()	150
<i>Sets a headline</i>	
Formular::setName()	150
<i>Sets a name</i>	
Formular::setStyle()	150
<i>Sets a style</i>	
Formular::updateDB()	151
<i>updates the database with the current values of the variables</i>	
Formular::setTextField()	151
<i>Modifies the textfield's text at postion \$position</i>	
Formular::setFormField()	149
<i>Modifies a formfields values, at postion \$position</i>	
Formular::setEnumeration()	149
<i>Modifies a enumeration by deleting the old one and adding a new one with the new values</i>	
Formular::getValues()	148
<i>Returns the values of this form</i>	
Formular::getSections()	148
<i>Returns the values of all sections of this form</i>	
Formular::load()	148
<i>loads the form with the ID from the SystemDB</i>	
Formular::moveSectionDown()	148
<i>Moves the section down</i>	
Formular::setDependency()	149
<i>Sets a dependency</i>	
Formular::moveSectionUp()	148
<i>Moves the section up</i>	
Formular::\$headline	145
Formular::\$dependency	145
functions_advanced.php	27
<i>A collection of functions that make use of the Database</i>	
functions.php	26
<i>Collection of frequently used Functions</i>	
form_dependency_exec.php	25
<i>Here just the selected form dependencies are stored in the DB</i>	
functions_copy_form.php	32
<i>Function to create a form out of another form</i>	
functions_help.php	34
<i>This file contains functions necessary for the context sensitve help</i>	
FileGenerator	142
<i>FileGenerator class contains functions to create, close and writing into a file</i>	
fieldError()	101
form_dependency.php	24
<i>Here the dependencies between search - report - change formulars can be set</i>	
formular_load.php	23

<i>This file loads a formular specified with its ID</i>	
formular_add_choose_formular.php	18
<i>This file asks for the concrete formular that is going to be copied.</i>	
formular_add_choose_exec.php	17
<i>This file performs the copying of a formular.</i>	
FileGenerator.php	16
<i>This file contains the classes: FileGenerator, HTMLGenerator and PHPGenerator.</i>	
formular_add_choose_project.php	19
<i>This file asks for the source of a formular during copying of formulars.</i>	
formular_create_check.php	20
<i>This file checks for existing and forbidden names of a formular.</i>	
formular_create_specify.php	22
<i>This file provides a HTML formular for creation of a new formular.</i>	
formular_create_exec.php	21
<i>This file performs the creation of a new formular..</i>	
FileGenerator::\$file_pointer	142
FileGenerator::closeFile()	142
<i>Closes a file</i>	
fileoutput::close()	144
<i>Closes the file</i>	
fileoutput::\$files	144
fileoutput	144
<i>Output data to textfiles</i>	
fileoutput::write()	144
<i>Write data to the file</i>	
fileoutput::writeln()	144
<i>Writes a line to the file</i>	
Formular::\$debug	145
Formular	145
<i>Contains all data of a formular</i>	
fileinput::readAll()	143
<i>Reads an entire file into an array.</i>	
fileinput::read()	143
<i>Reads a single line</i>	
FileGenerator::my_split()	142
<i>Splits a string into tokens at some specific points</i>	
FileGenerator::createFile()	142
<i>Creates and begins a file</i>	
FileGenerator::write()	142
<i>Writes text into the file</i>	
fileinput	143
<i>Loads data from textfiles</i>	
fileinput::close()	143
<i>Closes the file</i>	
fileinput::\$files	143
filefunctions.php	15
<i>This class provides functions to deal with files like reading or writing</i>	

G

generateStyles()	84
generatePreviewForms()	84

generateForms()	84
getFormStyle()	84
getSectionStyle()	84
getStyleNames()	108
getTablesAndFields()	101
getStyleName()	84
generateCSSFilename()	84
getFormType()	35
<i>Returns the formtype as an actual word</i>	
getFormsByType()	28
<i>Returns all forms of a given Type</i>	
getFieldType()	28
<i>Returns the type of the field form the target database</i>	
getFieldsForTable()	27
<i>Creates an drop down menu content with all fieldnames from a table</i>	
getFormsWithDependency()	28
<i>Returns forms of a given type whith a given dependency</i>	
getMenu()	28
<i>Creates an HTML menu from a database query</i>	
general_functions.php	35
<i>A collection of functions that don't need anything special</i>	
getPossibleTablesAfterJoin()	29
<i>Returns tables that are joined in the form</i>	
getPossibleAppearance()	29
<i>Returns the possible appearances array for a given fieldtype</i>	
getDependency()	27
<i>Returns the dependency of a form</i>	

H

HTMLGenerator::makeTableBegin()	155
<i>Creates opening tag for a table</i>	
HTMLGenerator::makeTableEnd()	155
<i>Creates finishing tag for a table</i>	
HTMLGenerator::makeTDBegin()	155
<i>Creates opening tag for a table cell</i>	
HTMLGenerator::makeSingleLineInputField()	155
<i>Creates single-line input-field</i>	
HTMLGenerator::makeReturnToIndexLink()	155
<i>Creates Link to the project index</i>	
HTMLGenerator::makeMultiLineInputField()	154
<i>Creates multi-line input-field</i>	
HTMLGenerator::makeRadioButton()	154
<i>Creates a radio button</i>	
HTMLGenerator::makeTDColspanBegin()	155
<i>Creates opening tag for a table cell ranging over several table celss</i>	
HTMLGenerator::makeTDEnd()	156
<i>Creates finishing tag for a table cell</i>	
HTMLGenerator::transmitChangedPrimaryKeyValue()	156
<i>Creates a table with result of changed primary keys after the change_id_formular has been used</i>	
HTMLGenerator::writehtml()	156

<i>Writes arbitrary string into the HTML document</i>	
HTMLGenerator::makeULLink()	156
<i>Creates a unordered list</i>	
HTMLGenerator::makeTREnd()	156
<i>Creates finishing tag for a table row</i>	
HTMLGenerator::makeTextField()	156
<i>Creates paragraph</i>	
HTMLGenerator::makeTRBegin()	156
<i>Creates opening tag table row</i>	
HTMLGenerator::makeLabel()	154
<i>Creates a label for a HTML formular element</i>	
HTMLGenerator::makeHTMLHead()	154
<i>Creates HTML header</i>	
HTMLGenerator::makeChangePrimaryKeyLink()	151
<i>Offers a link to change the primary key. If the primary key was changed, the value is displayed.</i>	
HTMLGenerator::makeCheckbox()	152
<i>Creates a checkbox</i>	
HTMLGenerator::makeBR()	151
<i>Creates a row break</i>	
HTMLGenerator	151
<i>HTMLGenerator class contains functions to create a HTML file.</i>	
headline_specify.php	38
<i>This file provides a HTML input field for the headline of a formular.</i>	
help_window.php	39
<i>This page enables the help popup with helptext</i>	
HTMLGenerator::makeDropDownMenu()	152
<i>Creates a DropDown menu</i>	
HTMLGenerator::makeEasyDropDownMenu()	152
<i>Creates a DropDown menu where option value and description are the same</i>	
HTMLGenerator::makeFormularButton()	153
<i>Creates a HTML formular button</i>	
HTMLGenerator::makeHTMLEnd()	153
<i>Finishes HTML document and adds end tags</i>	
HTMLGenerator::makeFormHead()	153
<i>Creates headline consisting of logo and title</i>	
HTMLGenerator::makeFormGetBegin()	153
<i>Creates beginning of a HTML formular (GET method)</i>	
HTMLGenerator::makeFormBegin()	152
<i>Creates beginning of a HTML formular (POST method)</i>	
HTMLGenerator::makeFormEnd()	153
<i>Finishes a HTML formular</i>	
headline_exec.php	37
<i>This file performs the setting of a headline for a formular.</i>	
<hr/>	
install_db_exec.php	45
<i>This file performs the installation and configuration of DBeasy and creating the first sysadmin account.</i>	
install_db_specify.php	46
<i>This file provides an interface for installation and configuration of DBeasy and creating the first sysadmin account.</i>	

ifEqual()	105
install_db_connection.php	44
<i>This file provides an interface for inserting the access data for the DBeasy system database depending on the database type the user has chosen before.</i>	
inputtochange_exec.php	43
<i>Actually creates the new form from the current one based on the changes selected before</i>	
index.php	40
<i>This file redirects to the login interface.</i>	
index_create.php	41
<i>This file creates a project index.</i>	
inputtochange.php	42
<i>The values from the current form will be copied into a new form of the current project</i>	
inputcheck()	14
<i>Checks on SQL Injection attacks (;, ', --)</i>	

J

join_specify_search.php	51
<i>Just a redirection to the right place for setting Join Conditions</i>	
join_specify.php	50
join_exec_single.php	49
<i>This file creates a join within a table</i>	
join_exec.php	48
<i>This file is responsible for inserting a join</i>	
join_delete.php	47
<i>This file is responsible for deleting a join</i>	

L

login_exec.php	57
<i>This file performs the login.</i>	
logo_exec.php	58
<i>This file performs the setting of a logo for a formular.</i>	
logo_specify.php	59
<i>This file provides a HTML input field for the logo path.</i>	
login.php	56
<i>This file provides the login interface.</i>	
layout_specify.php	55
<i>This file provides the possibility to set and change the layout of a formular.</i>	
layout_exec.php	52
<i>This file performs the setting of a layout for a formular.</i>	
layout_report_exec.php	53
<i>Adjust the layout for a report</i>	
layout_report_specify.php	54
<i>The layout for a report can be selected</i>	
loadHelpFile()	15

M

menu_formularadmin_project.php	68
--	----

<i>This file contains the menu for project administration.</i>	
menu_formularadmin_formular_suche.php	67
<i>This file contains the submenu for editing a search formular.</i>	
menu_formularadmin_formular_eintrag.php	66
<i>This file contains the submenu for editing a insert formular.</i>	
menu_sysadmin_manage_formularadmin.php	69
<i>This file contains the menu for Formadmin account administration.</i>	
menu_sysadmin_manage_sysadmin.php	70
<i>This file contains the menu for Sysadmin account administration.</i>	
makeInsertArray()	106
makeHTML()	94
<i>Prepares the HTML for a given query</i>	
menu_formularadmin_formular_bericht.php	65
<i>This file contains the submenu for editing a report.</i>	
menu_formularadmin_formular_aenderung.php	64
<i>This file contains the submenu for editing a change formular.</i>	
makeMenuMultyArray()	30
<i>Displays and drop down menu based on an multy array</i>	
makeMenu2()	29
<i>Displays an entire drop down menu built from a values array The displayed value will be submitted</i>	
my_dir_copy()	15
<i>Copies entire directories</i>	
makeMenuMultyArray2()	30
<i>Displays and drop down menu based on an multy array with additional parameters</i>	
mainmenu_formularadmin.php	60
<i>This file contains the mainmenu on Formadmin level.</i>	
menu_formularadmin_formular.php	63
<i>This file contains the menu for formular administration.</i>	
mainmenu_sysadmin.php	62
<i>This file contains the mainmenu on Sysadmin level.</i>	
my_copy()	15
<i>Copies files from a path to another path</i>	

O

odbc_connection_form.php	7
<i>This file displays input fields required to connect to odbc</i>	

P

PHPGenerator	157
<i>PHPGenerator class contains functions to create a executing PHP file.</i>	
PHPGenerator::makeAutoReturn()	157
PHPGenerator::makeChangeIDFormular()	157
<i>Creates the formular for changing the primary key</i>	
PHPGenerator::makeConnection()	157
<i>Creates connection statements to database</i>	
push_error()	101
publishForm()	94
<i>Publishes the actual form or displays a preview</i>	

publish_report_exec.php	88
<i>This file performs the publishing of a report.</i>	
publish_search.php	89
<i>This file introduces the publishing of a search formular.</i>	
publish_search_exec.php	90
<i>This file performs the creation of the search intelligence of the final forms</i>	
PHPGenerator::makeFormHead()	158
<i>Creates headline consisting of logo and title</i>	
PHPGenerator::makeHTMLHead()	158
<i>Creates HTML header</i>	
PHPGenerator::makeReturnToIndexLink()	159
<i>Creates Link to the project index</i>	
PHPGenerator::makeSQL_insert()	159
<i>Creates the executing PHP file for an Insert formular</i>	
PHPGenerator::makeSQL_update()	159
<i>Creates the executing PHP file for a Change formular</i>	
PHPGenerator::makeReturnButton()	159
<i>Creates a back link</i>	
PHPGenerator::makePHPend()	159
<i>Finishes a PHP file</i>	
PHPGenerator::makeIncludeAdo()	158
<i>Creates an include of ADOdb files</i>	
PHPGenerator::makeInputCheck()	158
<i>Introducing SQL injection test</i>	
PHPGenerator::makePHPBegin()	159
<i>Creates beginning of PHP file and adds comments about date and author of this PHP file</i>	
publish_report.php	87
<i>This file introduces the publishing of a report.</i>	
publish_eintrag_exec.php	86
<i>This file performs the publishing of a insert formular.</i>	
project_create_exec.php	75
<i>This file performs the creation of a project.</i>	
project_create_specify.php	76
<i>This file provides a HTML formular for creating a project.</i>	
project_delete.php	77
<i>This file performs the delete of a project.</i>	
project_delete_form.php	78
<i>This file lets you delete a whols Form</i>	
preview_suche.php	74
<i>This file generates and displays the preview of a search formular.</i>	
preview_report.php	73
<i>This file generates and displays the preview of a report.</i>	
previewLoad()	35
<i>Loads the given url to the main frame and updates the preview if it is activated</i>	
preview_aenderung.php	71
<i>This file generates and displays the preview of a change formular.</i>	
preview_eintrag.php	72
<i>This file generates and displays the preview of a insert formular.</i>	
project_delete_warn.php	79
<i>This file warns of the delete of a project.</i>	
project_edit.php	80
<i>This file offers the possibility to edit and delete a project.</i>	
publishFormCSS()	84

publishReportCSS()	84
publish_eintrag.php	85
<i>This file introduces the publishing of a insert formular.</i>	
publishCSS()	84
<i>return value: array containing name of CSS files</i>	
publish_css.php	84
<i>This file handles all CSS associated functions</i>	
project_show_formular.php	81
<i>This file performs the copying of a formular.</i>	
publish_aenderung.php	82
<i>This file introduces the publishing of a change formular.</i>	
publish_aenderung_exec.php	83
<i>This file performs the publishing of a change formular.</i>	
primaryIntKeyExists()	30
<i>Returns the primary Integer key of \$table - or false if none exists</i>	

R

report_fieldspecify.php	93
<i>The fields that should be displayed in a report can be selected here</i>	
report_functions.php	94
<i>A collection of functions that are used for publishing and displaying a report</i>	
report_specify.php	96
<i>Here addition conditions for a report can be specified</i>	
report_exec.php	92
<i>Previously specified conditions are actually inserted in this file or existing conditions can be deleted.</i>	
reportfield_exec.php	91
<i>The fields to be displays lateron within the report are written into the database</i>	
removeEntryFromArray()	31
<i>Removes an entry from a given array and returns this reduced array</i>	
reloadPreview()	35
<i>Updates preview if activated</i>	
reportIsEmpty()	73
removeDBeaslyTablesFromArray()	31
<i>Removes the dbeasly-tables from the given array and returns the reduced array</i>	

S

SQLFunctions::existsItem()	166
SQLFunctions::generateDate()	166
<i>Generates the Date, dependent on the databasetype</i>	
SQLFunctions::executeSQL()	166
<i>Executes the SQL-Statement given as an argument of the function</i>	
SQLFunctions::dropTable()	166
<i>Deletes a table from the database</i>	
SQLFunctions::dropColumn()	166
<i>Deletes the specified Column from the Table</i>	
SQLFunctions::generateID()	166
<i>Generates an unique ID</i>	
SQLFunctions::getFields()	167

	<i>Returns all fields of the specified table in an array</i>	
SQLFunctions::insertData()		167
	<i>Inserts an Datarecord into a table. \$value_array is an array which contains the new values.</i>	
SQLFunctions::getResultMsg()		167
	<i>Returns the Result Message</i>	
SQLFunctions::getMax()		167
	<i>Returns the maximal number of datarecords, which are in the table</i>	
SQLFunctions::getIntPrimarykey()		167
	<i>Returns the name of primary key, if there is only one primary key and primary key is of type integer, else false</i>	
SQLFunctions::deleteRow()		165
	<i>Deletes the row in a table which satisfies the given condition</i>	
SQLFunctions::createTable()		165
	<i>Creates a table which name and fields are given in the argument of the function</i>	
SQLFunctions::\$db_username		164
SQLFunctions::\$db_type		163
SQLFunctions::\$db_path		163
SQLFunctions::\$db_password		163
SQLFunctions::\$debug		164
SQLFunctions::addField()		164
	<i>Adds an field to the table mentioned in the arguments</i>	
SQLFunctions::connectToDBODBM()		165
	<i>Abbreviation: connectToDataBaseOverDataBaseMapping</i>	
SQLFunctions::connectToDB()		164
	<i>Connects to the Database specified in the arguments of the function</i>	
SQLFunctions::autoConnectToDB()		164
	<i>Connects to the Database specified in the file config_user.php</i>	
SQLFunctions::addUser()		164
	<i>Adds an user to the table dbeasy_user</i>	
SQLFunctions::LogTable()		168
	<i>Adds the tablename to the dbeasy_log_systables, needed for uninstallation</i>	
SQLFunctions::selectAll()		168
	<i>Returns all Datarecords of a table, which satisfy the given condition in an array.</i>	
SQLInstallFunctions::createStyleSectionsTable()		172
	<i>Creates DBeasy sytem table: StyleSection</i>	
SQLInstallFunctions::createStyleFormsTable()		171
	<i>Creates DBeasy sytem table: StyleForm</i>	
SQLInstallFunctions::createSectionTable()		171
	<i>Creates DBeasy sytem table: Section</i>	
SQLInstallFunctions::createReportFieldTable()		171
	<i>Creates DBeasy sytem table: ReportField</i>	
SQLInstallFunctions::createStyleStylesTable()		172
	<i>Creates DBeasy sytem table: StyleStyle</i>	
SQLInstallFunctions::createUserDBDataFile()		172
	<i>Creates a php file containing database access data for DBeasy's system tables</i>	
SQLInstallFunctions::createWorkfieldTable()		173
	<i>Creates DBeasy sytem table: Workfield</i>	
SQLInstallFunctions::createUserTable()		173
	<i>Creates DBeasy sytem table: User</i>	
SQLInstallFunctions::createUserDirMappingTable()		173
	<i>Creates DBeasy sytem table: UserDirMapping</i>	
SQLInstallFunctions::createUserDBMappingTable()		172
	<i>Creates DBeasy sytem table: UserDBMapping</i>	

SQLInstallFunctions::createReportCondTable()	171
<i>Creates DBeasy sytem table: ReportCondition</i>	
SQLInstallFunctions::createProjectTable()	171
<i>Creates DBeasy sytem table: Project</i>	
SQLInstallFunctions	169
<i>SQLInstallFunctions contains special functions for DBeasy installation concerning database issues.</i>	
SQLFunctions::updateRow()	169
<i>Updates an existing row, with the values in the variable \$fields_array</i>	
SQLFunctions::tableExists()	168
<i>Checks if the specified table exists in the database</i>	
SQLFunctions::showTables()	168
<i>Shows all tables, which are contained in the database to which the programm is connected</i>	
SQLInstallFunctions::createConditionTable()	169
<i>Creates DBeasy sytem table: Condition</i>	
SQLInstallFunctions::createFormularTable()	170
<i>Creates DBeasy sytem table: Formular</i>	
SQLInstallFunctions::createLogTable()	170
<i>Creates DBeasy sytem table: Logging</i>	
SQLInstallFunctions::createLogicTable()	170
<i>Creates DBeasy sytem table: Logic</i>	
SQLInstallFunctions::createJoinTable()	170
<i>Creates DBeasy sytem table: Join</i>	
SQLInstallFunctions::createHelpTable()	170
<i>Creates DBeasy sytem table: Help</i>	
SQLFunctions::\$db_name	163
SQLFunctions::\$db_dsn	163
style_index_process_1.php	109
<i>If you selected a new style in style index it is stored inside the form here.</i>	
style_index_process_2.php	110
style_index.php	108
<i>Style index contains the general management interface for styles.</i>	
style_editor_update.php	107
<i>just updates the database with the new values</i>	
style_editor_process.php	106
<i>This page just stores data in the db</i>	
style_index_process_3.php	111
<i>This file deletes a style from the database</i>	
Section	160
<i>A section is a part of a formular. Each section belongs to just one formular so it stores its formular ID.</i>	
Section::\$enumerations	160
Section::\$debug	160
Section::\$data_type	160
Section::\$appearance	160
style_editor.php	105
<i>Here all values for styles can be inserted</i>	
<i>Those values are passed on to style_editor_update or style_editor_process depending if a style is defined for the form or not.</i>	
start.php	104
<i>This is one of DBeasy's central files. Start.php manages most of the session variables and includes the right files into the frames depending on the context.</i>	
search_exec.php	97

<i>Collects all the necessary Data and then assembles the query</i>	
searchKeyInArray()	95
<i>Returns if a value appears in an array</i>	
showHelp()	34
<i>Shows the help icon with a link that pops up the help text of a category</i>	
standard_connection_form.php	9
<i>This file displays input fields required to connect to standard databases like mysql</i>	
section_management_exec.php	98
<i>Executes the rearranging, deleting and editing of sections (Formfield or textfields)</i>	
section_management_specify.php	99
<i>This file allows rearranging, deleting and editing of sections (Formfield or textfields)</i>	
SQLInstallFunctions.php	103
<i>SQLInstallFunctions contains special functions for DBeasy installation concerning database issues.</i>	
SQLFunctions.php	102
<i>This file contains functions concerning database connections and actions.</i>	
security_functions.php	101
<i>This file contains several functions concerning security of user actions.</i>	
security_check_logged_in.php	100
<i>This file checks for unauthorized access - meaning without previous, successful login - to a file.</i>	
Section::\$field	160
Section::\$form_id	160
Section::setPosition()	162
<i>Adjusts the position</i>	
Section::setFormField()	162
<i>Sets the parameters for a formfield</i>	
Section::loadSection()	162
<i>loads a section with a certain section ID</i>	
Section::loadEnumerations()	162
<i>Loads the enumeration for this section</i>	
Section::setTextField()	162
<i>Sets the parameter for a textfield</i>	
Section::updateDB()	163
<i>Updates the DB with the values of this section</i>	
SQLFunctions::\$db_address	163
SQLFunctions::\$current_result	163
SQLFunctions::\$connection	163
SQLFunctions	163
<i>Contains all functions which handle with SQL</i>	
Section::isText()	162
<i>Returns if this is a textfield</i>	
Section::getValues()	162
<i>Returns the values of this sections in an array with:</i>	
Section::\$position	161
Section::\$my_sql	161
Section::\$label	160
Section::\$id	160
Section::\$table	161
Section::\$text	161
Section::deleteSection()	161
<i>Deletes a section together with its enumeration</i>	
Section::deleteEnumeration()	161
<i>deletes the enumeration of this section</i>	

Section::createSection()	161
<i>Creates a section for a certain form at a certain position</i>	
Section::addEnumeration()	161
<i>Adds enumerations to this section, creates table if necessary</i>	
sqlite_connection_form.php	8
<i>This file displays input fields required to connect to sqlite</i>	

T

table_drop_column.php	119
<i>This file performs the delete of a table field.</i>	
table_drop.php	118
<i>This file performs the delete of a table.</i>	
table_drop_column_warn.php	120
<i>This file warns of the delete of a table field.</i>	
table_drop_warn.php	121
<i>This file warns of the delete of a table.</i>	
template.php	123
<i>This file is a template file for all kind of user interface files .</i>	
table_show_structure.php	122
<i>This file provides a HTML formular for editing and deleting a table fields.</i>	
table_create_specify.php	117
<i>This file provides a HTML formular for creating a table in a database.</i>	
table_create_exec.php	116
<i>This file performs the creation of a table in a database.</i>	
tableError()	101
table_tr()	99
table_add_field_exec.php	112
<i>This file performs the insert of a new table field into a table.</i>	
table_add_field_specify.php	113
<i>This file provides a HTML formular for inserting a new table field into a table.</i>	
table_create.php	115
<i>This file provides a HTML formular for creating a table in a database.</i>	
table_add_field_warn.php	114
<i>This file warns of the insert a new table field into a table.</i>	
translateFormType()	26

U

user_dir_mapping_exec.php	136
<i>This file performs the association of a directory with an user.</i>	
user_dir_mapping_delete.php	135
<i>This file performs the delete of an association of a directory with an user.</i>	
user_dir_mapping.php	134
<i>This file provides a HTML formular for association of a directory with an user.</i>	
user_dir_mapping_warn.php	137
<i>This file warns the user of the delete of an association of a directory with an user.</i>	
user_entry_exec.php	138
<i>This file performs the creation of an user account.</i>	
user_show_table.php	140
<i>This file provides a HTML formular for editing and deleting an user account.</i>	

user_entry_specify.php	139
<i>This file provides a HTML formular for creating an user account.</i>	
user_delete_warn.php	133
<i>This file warns the user of the delete of an user account.</i>	
user_delete.php	132
<i>This file performs the delete of an user account.</i>	
user_db_mapping_connection.php	127
<i>This file includes - depending of the database type - a special HTML formular where the access data can be inserted.</i>	
user_db_mapping.php	126
<i>This file provides a HTML formular for association of a database with an user.</i>	
uninstall_exec.php	125
<i>This file performs the deinstallation of DBeasy.</i>	
user_db_mapping_delete.php	128
<i>This file performs the delete of an association of a database with an user.</i>	
user_db_mapping_exec.php	129
<i>This file performs the association of a database with an user.</i>	
user_db_mapping_warn.php	131
<i>This file warns the user of the delete of an association of a database with an user.</i>	
user_db_mapping_specify.php	130
<i>This file provides a HTML formular for choosing the database type during the step of association of a database with an user.</i>	
uninstall.php	124
<i>This file warns of the deinstallation of DBeasy.</i>	

W

write2DimStringArray()	95
<i>Returns a string that can be used to write an 2 dimensional array with its values into a phpfile for later usage</i>	
write1DimStringArray()	95
<i>Returns a string that can be used to write an the array with its values into a phpfile for later usage</i>	