dbdMVC

Powered by phpDocumentor

Contents

Package dbdCommon Procedural Elements . . . 2
FlashLoaderSWFO.php . . . 2
GDColor.php . . . 3
InputCSS.php . . . 4
QTLoader.php . . . 5
wmArrays.php . . . 6
wmFileSystem.php . . . 7
wmSort.php . . . 8
wmString.php . . . 9
Package dbdCommon Classes . . . 10
Class FlashLoaderSWFO . . . 10
Var $_atts . . . 11
Var $_class_id . . . 11
Var $_codebase . . . 11
Var $_common_js_shown . . . 11
Var $_js_path . . . 12
Var $_movies . . . 12
Var $_pluginspage . . . 12
Var $_req_atts . . . 13
Var $_swfo_atts . . . 13
Constructor __construct . . . 13
Method addMovie . . . 14
Method getCommonJS . . . 14
Method getCommonJSPath . . . 14
Method getMovie . . . 14
Method getMovies . . . 15
Method hasMovies . . . 15
Method loadMovie . . . 15
Method loadMovies . . . 16
Method setAtts . . . 16
Method _buildSWFObject . . . 16
Method _getAtts . . . 16
Method _missingReqAtt . . . 17
Method _optionalAtt . . . 17
Class GDColor . . . 18
Var $color . . . 18
Var $im . . . 18
Var $rgba . . . 18
Constructor __construct . . . 19
Method getAlpha . . . 19
Method getBlue . . . 19
Method getColor . . . 19
Method getGreen . . . 20

Method getHex 20
Method getRed 20
Method hex2arr 20
Method int2arr 21
Method __destruct 21
Class InputCSS 21
Var $div_id 22
Var $input_ids 22
Method button 22
Method checkbox 23
Method file 23
Method getCallingFile 24
Method getDivID 24
Method getInputID 24
Method getXHTML 24
Method radio 25
Method reset 25
Class QTLoader 26
Var $atts 27
Var $class_id 27
Var $codebase 27
Var $common_js_shown 27
Var $js_path 28
Var $movies 28
Var $onload 28
Var $pluginspage 28
Var $req_atts 29
Constructor __construct 29
Method addMovie 29
Method buildQTObject 30
Method getAtts 30
Method getCommonJS 30
Method getCommonJSPath 31
Method getMovie 31
Method getMovies 31
Method hasMovies 31
Method loadMovie 32
Method loadMovies 32
Method missingReqAtt 32
Method optionalAtt 33
Method setAtts 33
Class wmArrays 33
Method arrayMergeRecursiveSumFields 34
Method arrayRemoveValue 34
Method arraySearchRecursive 35
Method arrayValuesRecursive 35
Method explodeNewline 35
Method implodeObjArray 36
Method pregExplode 36

Class wmFileSystem . . . 36
- Method bytesToHuman . . . 37
- Method dirSize . . . 37
- Method mimeType . . . 37
- Method scanDir . . . 38

Class wmSort . . . 38
- Var $sortby . . . 39
- Method multiASort . . . 39
- Method multiSort . . . 39
- Method _getVal . . . 40
- Method _wmMultiSort . . . 40

Class wmString . . . 41
- Class Constant TIME_LEN_FORMAT_LONG . . . 41
- Class Constant TIME_LEN_FORMAT_SHORT . . . 41
- Class Constant TIME_LEN_LABEL_AGO . . . 41
- Class Constant TIME_LEN_LABEL_DAY . . . 41
- Class Constant TIME_LEN_LABEL_HRS . . . 41
- Class Constant TIME_LEN_LABEL_LAST . . . 41
- Class Constant TIME_LEN_LABEL_MIN . . . 42
- Class Constant TIME_LEN_LABEL_MON . . . 42
- Class Constant TIME_LEN_LABEL_SEC . . . 42
- Class Constant TIME_LEN_LABEL_TODAY . . . 42
- Class Constant TIME_LEN_LABEL_WEEK . . . 42
- Class Constant TIME_LEN_LABEL_YEAR . . . 42
- Class Constant TIME_LEN_LABEL_YESTERDAY . . . 42
- Class Constant TIME_LEN_TIME_DAY . . . 42
- Class Constant TIME_LEN_TIME_HRS . . . 42
- Class Constant TIME_LEN_TIME_MIN . . . 42
- Class Constant TIME_LEN_TIME_MON . . . 42
- Class Constant TIME_LEN_TIME_SEC . . . 42
- Class Constant TIME_LEN_TIME_WEEK . . . 42
- Class Constant TIME_LEN_TIME_YEAR . . . 42
- Class Constant TIME_LEN_TYPE_DAY . . . 42
- Class Constant TIME_LEN_TYPE_HRS . . . 43
- Class Constant TIME_LEN_TYPE_MIN . . . 43
- Class Constant TIME_LEN_TYPE_MON . . . 43
- Class Constant TIME_LEN_TYPE_SEC . . . 43
- Class Constant TIME_LEN_TYPE_WEEK . . . 43
- Class Constant TIME_LEN_TYPE_YEAR . . . 43
- Class Constant VALID_DATE . . . 43
- Class Constant VALID_EMAIL . . . 43
- Var $number_suffixes . . . 43
- Var $plural_exceptions . . . 43
- Var $random_word_cons . . . 44
- Var $random_word_vowl . . . 44
- Method dateValid . . . 44
- Method dayOfWeek . . . 44
- Method emailEncode . . . 45
- Method emailValid . . . 45

Method hanukkahStart . . . 46
Method jsSpecialChars . . . 46
Method lcFirst . . . 46
Method makePlural . . . 47
Method makeSingular . . . 47
Method numberSuffix . . . 47
Method randomWord . . . 48
Method superCap2UnderLower . . . 48
Method superCapAddSpaces . . . 48
Method timeFormat . . . 49
Method timeLengthFormat . . . 49
Method timePastFormat . . . 50
Method truncate . . . 50
Method underLower2SuperCap . . . 50
Method wkt2kml . . . 51
Package FirePHP Procedural Elements . . . 53
FirePHP.class.php . . . 53
Package FirePHP Classes . . . 55
Class FirePHP . . . 55
Class Constant DUMP . . . 55
Class Constant ERROR . . . 55
Class Constant EXCEPTION . . . 56
Class Constant GROUP_END . . . 56
Class Constant GROUP_START . . . 56
Class Constant INFO . . . 56
Class Constant LOG . . . 56
Class Constant TABLE . . . 56
Class Constant TRACE . . . 56
Class Constant VERSION . . . 57
Class Constant WARN . . . 57
Var $convertAssertionErrorsToExceptions . . . 57
Var $enabled . . . 57
Var $inExceptionHandler . . . 57
Var $instance . . . 58
Var $json_objectStack . . . 58
Var $messageIndex . . . 58
Var $objectFilters . . . 59
Var $objectStack . . . 59
Var $options . . . 59
Var $throwAssertionExceptions . . . 59
Var $throwErrorExceptions . . . 60
Constructor __construct . . . 60
Method assertionHandler . . . 60
Method detectClientExtension . . . 60
Method dump . . . 61
Method encodeObject . . . 61
Method encodeTable . . . 61
Method error . . . 62

Method errorHandler . . . 62
Method exceptionHandler . . . 63
Method fb . . . 63
Method getEnabled . . . 63
Method getInstance . . . 64
Method getOptions . . . 64
Method getUserAgent . . . 64
Method group . . . 64
Method groupEnd . . . 65
Method info . . . 65
Method init . . . 65
Method is_utf8 . . . 66
Method jsonEncode . . . 66
Method json_encode . . . 66
Method json_name_value . . . 67
Method json_utf82utf16 . . . 67
Method log . . . 68
Method newException . . . 68
Method registerAssertionHandler . . . 68
Method registerErrorHandler . . . 69
Method registerExceptionHandler . . . 69
Method setEnabled . . . 69
Method setHeader . . . 70
Method setObjectFilter . . . 70
Method setOptions . . . 70
Method setProcessorUrl . . . 71
Method setRendererUrl . . . 71
Method table . . . 72
Method trace . . . 72
Method warn . . . 72
Method _escapeTrace . . . 73
Method _escapeTraceFile . . . 73
Method _standardizePath . . . 73
Method __sleep . . . 74
Package JSMin Classes . . . 75
Class JSMin . . . 75
Class Constant ORD_LF . . . 76
Class Constant ORD_SPACE . . . 76
Var $a . . . 76
Var $b . . . 76
Var $input . . . 76
Var $inputIndex . . . 77
Var $inputLength . . . 77
Var $lookAhead . . . 77
Var $output . . . 77
Constructor __construct . . . 77
Method action . . . 78
Method get . . . 78
Method isAlphaNum . . . 78

Method min . . . 78
Method minify . . . 79
Method next . . . 79
Method peek . . . 79
Class JSMinException . . . 79
Package dbdMVC Procedural Elements . . . 81
dbdMVC.php . . . 81
Function dbdLog . . . 81
Define DBD_CACHE_DIR . . . 82
Define DBD_DEBUG_ALL . . . 82
Define DBD_DEBUG_CSS . . . 82
Define DBD_DEBUG_DB . . . 82
Define DBD_DEBUG_HTML . . . 82
Define DBD_DEBUG_JS . . . 82
Define DBD_DS . . . 82
Define DBD_FIREPHP_DIR . . . 82
Define DBD_MVC_CLI . . . 82
Define DBD_PLUG_DIR . . . 82
Define DBD_SMARTY_DIR . . . 82
Define DBD_SMARTY_PLUG_DIR . . . 82
Define DBD_XAJAX_DIR . . . 82
fb.php . . . 84
Function fb . . . 84
dbconn.inc . . . 85
Define DBCONN_DEFAULT_DB . . . 85
Define DBCONN_INCLUDED . . . 85
Define DB_DRIVER . . . 85
Define DB_HOST . . . 85
Define DB_PASS . . . 85
Define DB_PORT . . . 85
Define DB_USER . . . 85
Index.php . . . 86
index.php . . . 87
Define DBD_APP_DIR . . . 87
Define DBD_DOC_ROOT . . . 87
Define DBD_MVC_DIR . . . 87
dbdController.php . . . 88
dbdCSS.php . . . 89
dbdDB.php . . . 90
dbdDispatcher.php . . . 91
dbdEmptyController.php . . . 92
dbdError.php . . . 93
dbdException.php . . . 94
dbdHoldableException.php . . . 95
dbdInfo.php . . . 96
dbdJS.php . . . 97
dbdLoader.php . . . 98
Function __autoload . . . 98
dbdModel.php . . . 99

dbdRequest.php . . . 100
dbdRouter.php . . . 101
dbdSmarty.php . . . 102
dbdURI.php . . . 103
dbdXajax.php . . . 104
dbdXajaxJS.php . . . 105
wmGD.php . . . 106
Function colorizeImage . . . 106
Function cropImage . . . 106
Function cropImageReal . . . 106
Function cropResizeImage . . . 106
Function imageCreateFrom . . . 106
Function imagefilledellipsealpha . . . 107
Function imageLineThick . . . 107
Function imageType . . . 107
Function imageTypesArray . . . 107
Function outputAndDestroy . . . 107
Function reflectImage . . . 107
Function resizeImage . . . 108
Function stretchCopyButton . . . 108
Package dbdMVC Classes . . . 109
Class dbdController . . . 109
Var $app_dir . . . 109
Var $default_render . . . 110
Var $model . . . 110
Var $router . . . 110
Var $template . . . 110
Var $view . . . 111
Var $xajax . . . 111
Var $xajaxResponse . . . 111
Constructor __construct . . . 112
Method assignAllParams . . . 112
Method autoExec . . . 112
Method autoRender . . . 113
Method dnit . . . 113
Method doDefault . . . 113
Method doXajax . . . 113
Method forward . . . 114
Method getAction . . . 114
Method getAppDir . . . 114
Method getController . . . 114
Method getParam . . . 115
Method getParams . . . 115
Method getTemplate . . . 115
Method getUploadInfo . . . 115
Method getURL . . . 116
Method init . . . 116
Method noRender . . . 117
Method registerXajaxActions . . . 117

Method setParam . . . 117
Method setTemplate . . . 118
Method unsetParam . . . 118
Method __destruct . . . 118
Class dbdCSS . . . 119
Class Constant BLOCK_CLOSE_REGEX . . . 119
Class Constant BLOCK_OPEN_REGEX . . . 119
Class Constant BUTT_DEF_TAG_REGEX . . . 119
Class Constant BUTT_GEN_TAG_REGEX . . . 120
Class Constant BUTT_GEN_TPL_TAG_REGEX . . . 120
Class Constant BUTT_PROP_DEF_DISABLED_REGEX . . . 120
Class Constant BUTT_PROP_DEF_HOVER_REGEX . . . 120
Class Constant BUTT_PROP_DEF_REGEX . . . 120
Class Constant BUTT_TAG_REGEX . . . 120
Class Constant CACHE_INFO_FILES . . . 120
Class Constant CACHE_INFO_SPRITE . . . 121
Class Constant CSS_EXT_REGEX . . . 121
Class Constant DEBUG_REGEX . . . 121
Class Constant DIR_DELIM_REGEX . . . 121
Class Constant IMPORT_REGEX . . . 121
Class Constant MATH_CALL_REGEX . . . 121
Class Constant MATH_SAFE_REGEX . . . 121
Class Constant VAR_CALL_REGEX . . . 122
Class Constant VAR_DEF_REGEX . . . 122
Class Constant VAR_TAG_REGEX . . . 122
Var $buffer . . . 122
Var $buttons . . . 122
Var $buttons_generated . . . 123
Var $button_props . . . 123
Var $button_props_generated_global . . . 123
Var $button_props_generated_tpl . . . 124
Var $button_props_global . . . 124
Var $button_props_required . . . 124
Var $cache_dir . . . 124
Var $cache_file . . . 125
Var $debug . . . 125
Var $file . . . 125
Var $files . . . 126
Var $line . . . 126
Var $line_num . . . 126
Var $minify_regex . . . 126
Var $sprite . . . 127
Var $vars . . . 127
Method checkButtonCache . . . 127
Method checkButtonPropsRequired . . . 128
Method checkCache . . . 128
Method createCache . . . 128
Method doCombine . . . 128
Method dumpFile . . . 129

Method ensureResource . . . 129
Method genButton . . . 129
Method genButtonCacheFile . . . 130
Method genButtonCSS . . . 130
Method genButtons . . . 130
Method genButtonSprite . . . 131
Method genCacheName . . . 131
Method genURL . . . 131
Method getButtonProps . . . 132
Method getVar . . . 132
Method init . . . 132
Method math . . . 132
Method minify . . . 133
Method output . . . 133
Method parseImports . . . 133
Method setHeaders . . . 134
Class dbdDB . . . 134
Class Constant COMP_BETWEEN . . . 134
Class Constant COMP_EQ . . . 134
Class Constant COMP_GT . . . 134
Class Constant COMP_GTEQ . . . 134
Class Constant COMP_IN . . . 135
Class Constant COMP_LIKE . . . 135
Class Constant COMP_LT . . . 135
Class Constant COMP_LTEQ . . . 135
Class Constant COMP_NBETWEEN . . . 135
Class Constant COMP_NEQ . . . 135
Class Constant COMP_NIN . . . 135
Class Constant COMP_NLIKE . . . 135
Class Constant COMP_NULLEQ . . . 135
Class Constant COMP_TYPE . . . 135
Class Constant DATE_FORMAT . . . 135
Class Constant DBCONN_PATH . . . 135
Class Constant TIME_FORMAT . . . 136
Var $instances . . . 136
Var $options . . . 136
Var $pdo . . . 137
Constructor __construct . . . 137
Method assembleDSN . . . 137
Method date . . . 138
Method getInstance . . . 138
Method includeDBConn . . . 139
Method nextAutoID . . . 139
Method prepExec . . . 139
Method setAttributes . . . 140
Method time . . . 140
Method __call . . . 140
Class dbdDispatcher . . . 141
Class Constant ACTION_PREFIX . . . 141

- Class Constant BUILTIN_PREFIX . . . 141
- Class Constant DEFAULT_ACTION . . . 141
- Class Constant DEFAULT_CONTROLLER . . . 141
- Class Constant ERROR_CONTROLLER . . . 142
- Class Constant FALLBACK_CONTROLLER . . . 142
- Class Constant MAGIC_ACTION . . . 142
- Class Constant XAJAX_ACTION_PREFIX . . . 142
- Var $app_dir . . . 142
- Var $controller_dirs . . . 142
- Var $fallback_controller . . . 143
- Var $router . . . 143
- Constructor __construct . . . 143
- Method dispatch . . . 143
- Method getAction . . . 144
- Method getController . . . 144

Class dbdEmptyController . . . 144

Class dbdError . . . 145
- Class Constant ERR_400 . . . 145
- Class Constant ERR_401 . . . 145
- Class Constant ERR_403 . . . 145
- Class Constant ERR_404 . . . 145
- Class Constant ERR_408 . . . 146
- Class Constant ERR_500 . . . 146
- Method doDefault . . . 146
- Method doError . . . 146

Class dbdException . . . 147
- Constructor __construct . . . 147

Class dbdHoldableException . . . 147
- Var $held . . . 148
- Var $hold . . . 148
- Method getHeld . . . 148
- Method hold . . . 148
- Method intercept . . . 149
- Method release . . . 149

Class dbdInfo . . . 149
- Method doDefault . . . 150

Class dbdJS . . . 150
- Class Constant DIR_DELIM_REGEX . . . 150
- Var $buffer . . . 150
- Var $cache_dir . . . 151
- Var $cache_file . . . 151
- Var $debug . . . 151
- Var $files . . . 152
- Var $vars . . . 152
- Method addFile . . . 152
- Method addVars . . . 152
- Method checkCache . . . 153
- Method createCache . . . 153
- Method doCombine . . . 153

Method doDefault . . . 153
Method doGet . . . 153
Method ensureResource . . . 154
Method genCacheName . . . 154
Method genURL . . . 154
Method init . . . 155
Method minify . . . 155
Method output . . . 155
Method readFiles . . . 155
Method setHeaders . . . 156
Class dbdLoader . . . 156
Class Constant DEFAULT_CONTROLLER_DIR . . . 156
Class Constant DEFAULT_MODEL_DIR . . . 156
Var $controller_dirs . . . 157
Var $include_paths . . . 157
Var $instance . . . 157
Var $model_dir . . . 157
Constructor __construct . . . 158
Method addControllerDir . . . 158
Method addPath . . . 158
Method getControllerDirs . . . 159
Method getInstance . . . 159
Method getModelDir . . . 159
Method getPaths . . . 159
Method init . . . 160
Method load . . . 160
Method loadClass . . . 160
Method register . . . 161
Method search . . . 161
Method setControllerDir . . . 161
Method setModelDir . . . 162
Class dbdModel . . . 162
Class Constant CONST_TABLE_KEY . . . 163
Class Constant CONST_TABLE_NAME . . . 163
Var $data . . . 163
Var $db . . . 163
Var $id . . . 163
Var $reflections . . . 164
Var $table_key . . . 164
Var $table_name . . . 164
Constructor __construct . . . 164
Method buildWhereClause . . . 165
Method delete . . . 165
Method ensureDB . . . 165
Method getAll . . . 166
Method getConstant . . . 166
Method getCount . . . 166
Method getData . . . 167
Method getDB . . . 167

- Method getID . . . 167
- Method getReflection . . . 168
- Method init . . . 168
- Method initFields . . . 168
- Method save . . . 168
- Method setDB . . . 169
- Method __call . . . 169
- Method __get . . . 169
- Method __isset . . . 170
- Method __set . . . 170
- Method __unset . . . 170

Class dbdMVC . . . 171

- Class Constant VERSION . . . 171
- Var $app_dir . . . 171
- Var $app_name . . . 172
- Var $debug_mode . . . 172
- Var $dispatcher . . . 172
- Var $error_controller . . . 173
- Var $error_log . . . 173
- Var $expose_php_info . . . 173
- Var $fallback_controller . . . 173
- Var $instance . . . 174
- Var $request . . . 174
- Var $router . . . 174
- Var $start . . . 175
- Constructor __construct . . . 175
- Method debugMode . . . 175
- Method dispatch . . . 175
- Method e . . . 176
- Method exposePHPInfo . . . 176
- Method genCLIURI . . . 176
- Method getAppDir . . . 177
- Method getAppName . . . 177
- Method getErrorController . . . 177
- Method getErrorLog . . . 177
- Method getExecutionTime . . . 178
- Method getFallbackController . . . 178
- Method getInstance . . . 178
- Method getRequest . . . 178
- Method getRouter . . . 178
- Method init . . . 179
- Method logExecutionTime . . . 179
- Method PHPisExposed . . . 179
- Method run . . . 179
- Method setAppDir . . . 180
- Method setAppName . . . 180
- Method setDebugMode . . . 181
- Method setErrorController . . . 181
- Method setErrorLog . . . 181

- Method setFallbackController . . . 182
Class dbdRequest . . . 182
- Var $params . . . 183
- Var $request_uri . . . 183
- Constructor __construct . . . 183
- Method get . . . 183
- Method getCookie . . . 184
- Method getEnv . . . 184
- Method getHeader . . . 185
- Method getPost . . . 185
- Method getQuery . . . 186
- Method getRequestURI . . . 186
- Method getServer . . . 186
- Method has . . . 187
- Method isXmlHttpRequest . . . 187
- Method set . . . 187
- Method setRequestURI . . . 188
- Method __isset . . . 188

Class dbdRouter . . . 188
- Var $action . . . 189
- Var $baseUrl . . . 189
- Var $controller . . . 189
- Var $params . . . 190
- Var $request . . . 190
- Constructor __construct . . . 190
- Method addParam . . . 190
- Method buildURL . . . 191
- Method getAction . . . 191
- Method getController . . . 191
- Method getParam . . . 191
- Method getParams . . . 192
- Method getURL . . . 192
- Method parseRequest . . . 192
- Method setAction . . . 193
- Method setController . . . 193
- Method setParam . . . 193
- Method unsetParam . . . 194

Class dbdSmarty . . . 194
- Class Constant CACHE_DIR . . . 194
- Class Constant COMPILED_TEMPLATE_DIR . . . 195
- Class Constant CONFIG_DIR . . . 195
- Class Constant TEMPLATE_DIR . . . 195
- Class Constant VIEW_DIR . . . 195
- Var $app_dir . . . 195
- Var $beautify_regex . . . 195
- Var $css_files . . . 196
- Var $css_host . . . 196
- Var $css_vars . . . 196
- Var $debug . . . 196

- Var $flash_loader 197
- Var $js_files 197
- Var $js_host 197
- Var $js_vars 197
- Var $minify_regex 198
- Var $qt_loader 198
- Var $rendered 198
- Constructor __construct 199
- Method addCSS 199
- Method addCSSVar 199
- Method addFlashMovie 200
- Method addJS 200
- Method addJSVar 200
- Method addQTMovie 201
- Method clearCSS 201
- Method clearJS 201
- Method dbdBeautify 201
- Method dbdIncludeFiles 202
- Method dbdMinify 202
- Method dbdTag 203
- Method display 203
- Method fetch 203
- Method getCSSVar 204
- Method getJSVar 204
- Method initFlash 205
- Method initQuickTime 205
- Method setCssHost 205
- Method setJSHost 206
- Method setOptions 206
- Method wasRendered 206

Class dbdURI 206
- Var $action 207
- Var $controller 207
- Var $instance 207
- Var $params 208
- Var $position_lists 208
- Method create 208
- Method getAction 208
- Method getController 209
- Method getInstance 209
- Method getParams 209
- Method replace 209
- Method set 210
- Method setParam 210
- Method setPositionList 211

Class dbdXajax 211
- Var $router 211
- Constructor __construct 212
- Method _xmlToArray 212

Class dbdXajaxJS . . . 213
- Method doDefault . . . 213

Class FB . . . 213
- Method dump . . . 213
- Method error . . . 214
- Method getEnabled . . . 214
- Method getOptions . . . 214
- Method group . . . 215
- Method groupEnd . . . 215
- Method info . . . 215
- Method log . . . 216
- Method send . . . 216
- Method setEnabled . . . 217
- Method setObjectFilter . . . 217
- Method setOptions . . . 217
- Method table . . . 218
- Method trace . . . 218
- Method warn . . . 219

Class Index . . . 219
- Method doDefault . . . 219

Class wavEdit . . . 220
- Var $afiles . . . 220
- Var $arrData . . . 220
- Var $bdColor . . . 220
- Var $bgColor . . . 220
- Var $Db . . . 220
- Var $gridColor . . . 220
- Var $grpColor . . . 220
- Var $header . . . 220
- Var $imgBorder . . . 220
- Var $imgHeight . . . 220
- Var $imgWidth . . . 221
- Var $visual_graph_mode . . . 221
- Var $wavs . . . 221
- Constructor wavEdit . . . 221
- Method cropFiles . . . 221
- Method doClip . . . 221
- Method doCrop . . . 221
- Method doDrawImg . . . 221
- Method doEcho . . . 221
- Method doFade . . . 222
- Method doJoin . . . 222
- Method doSetVol8Bit . . . 222
- Method doSetVol16Bit . . . 222
- Method doViewHeader . . . 222
- Method drawImgWave . . . 222
- Method getBytes . . . 222
- Method getDataArr . . . 222
- Method getFileName . . . 223

Method getFiles . . . 223
Method getHeader . . . 223
Method getStructure . . . 223
Method joinFiles . . . 223
Method setEcho . . . 223
Method setFade . . . 223
Method setVolume . . . 224
Method unpackData . . . 224
Method unpackHeader . . . 224
Method viewHeader . . . 224
Method viewImage . . . 224
Method writeWavFile . . . 224
Appendices . . . 225
Appendix A - Class Trees . . . 226
dbdCommon . . . 226
dbdMVC . . . 227
JSMin . . . 229
FirePHP . . . 229
Appendix D - Todo List . . . 231

Package dbdCommon Procedural Elements

FlashLoaderSWFO.php

FlashLoaderSWFO.php :: Flash movie loading/embedding class

- **Package** dbdCommon
- **Author** Will Mason < will@dontblinkdesign.com>
- **Version** 1.3
- **Copyright** Copyright (c) 2006-2007 by Don't Blink Design

GDColor.php

GDColor.php :: The GD Color Class

GDColor version 1.1 Copyright (c) 2006-2007 by Don't Blink Design http://www.dontblinkdesign.com

- **Package** dbdCommon
- **Author** Will Mason < will@dontblinkdesign.com>
- **Version** 1.1
- **Copyright** Copyright (c) 2007 by Don't Blink Design
- **License** http://www.gnu.org/copyleft/lesser.html

InputCSS.php

InputCSS.php :: The Fancy XHTML/CSS Form Input Class

- **Package** dbdCommon
- **Author** Will Mason < will@dontblinkdesign.com>
- **Version** 1.3
- **Copyright** Copyright (c) 2009 by Don't Blink Design

QTLoader.php

QTLoader.php :: QuickTime movie loading/embedding class

- **Package** dbdCommon
- **Author** Will Mason < will@dontblinkdesign.com>
- **Version** 1.1
- **Copyright** Copyright (c) 2006-2007 by Don't Blink Design

wmArrays.php

wmArrays.php :: Will Mason's array processing class

- **Package** dbdCommon
- **Author** Will Mason < will@dontblinkdesign.com>
- **Version** 1.0
- **Copyright** Copyright (c) 2006-2007 by Don't Blink Design

wmFileSystem.php

wmFileSystem.php :: Will Mason's file system class

- **Package** dbdCommon
- **Author** Will Mason < will@dontblinkdesign.com>
- **Version** 1.2
- **Copyright** Copyright (c) 2006-2007 by Don't Blink Design

require_once **"wmSort.php"** *[line 10]*

wmSort.php

wmSort.php :: Will Mason's array sorting class

- **Package** dbdCommon
- **Author** Will Mason < will@dontblinkdesign.com>
- **Version** 1.1
- **Copyright** Copyright (c) 2006-2007 by Don't Blink Design

wmString.php

wmString.php :: Will Mason's string manipulation class

- **Package** dbdCommon
- **Author** Will Mason < will@dontblinkdesign.com>
- **Version** 1.3
- **Copyright** Copyright (c) 2006-2007 by Don't Blink Design

Package dbdCommon Classes

Class FlashLoaderSWFO

[line 28]

The FlashLoaderSWFO class is used to effortlessly embedd Flash movies in a (X)HTML Document.
REQUIRED ITEM: SWFObject.js

Simple Usage:

```
$FL = new FlashLoaderSWFO("version=6,0,29,0"          , "quality=high"         );
$FL->  loadMovie("src=swfs/movie.swf"          , "element_id=mainFlash"          ,
"width=800"          , "height=600"          );
```

Advanced Usage:

```
$FL = new FlashLoaderSWFO("version=6,0,29,0"          , "quality=high"         );
$FL->  addMovie("src=swfs/movie.swf"          , "element_id=mainFlash"          ,
"width=800"          , "height=600"          );
$FL->  loadMovies();
```

- **Package** dbdCommon

FlashLoaderSWFO::$_atts

array = array(
"useexpressinstall" => false,
"swliveconnect" => false,
"allowfullscreen" => false,
"play" => false,
"loop" => false,
"menu" => false,
"quality" => false,
"scale" => false,
"align" => false,
"salign" => false,

"wmode" => false,
"bgcolor" => false,
"base" => false,
"flashvars" => false,
"allowscriptaccess" => false
) *[line 93]*

list of main attribute defaults

- **Link** http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn_12701
- **Access** private

FlashLoaderSWFO::$_class_id

string = "D27CDB6E-AE6D-11cf-96B8-444553540000" *[line 43]*

the flash plugin class id

- **Access** private

FlashLoaderSWFO::$_codebase

string = "http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=" *[line 48]*

codebase without version number

- **Access** private

FlashLoaderSWFO::$_common_js_shown

boolean = false *[line 58]*

flag to be set so we write JavaScript only once

- **Access** private

FlashLoaderSWFO::$_js_path

string = "common_js/swfobject.js" *[line 38]*

path to SWFObject external JavaScripts library

- **Link** http://blog.deconcept.com/swfobject/
- **Access** private

FlashLoaderSWFO::$_movies

array = array() *[line 64]*

list of movies added my addMovie()

- **See** function addMovie
- **Access** private

FlashLoaderSWFO::$_pluginspage

string = "http://www.macromedia.com/go/getflashplayer" *[line 53]*

plugins page url

- **Access** private

FlashLoaderSWFO::$_req_atts

```
array = array(
        "src" => false,
        "width" => false,
        "height" => false,
        "version" => false,
        "element_id" => false,
        "id" => 1
        ) [line 70]
```

list of required attribute defaults

- **Link** http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn_12701
- **Access** private

FlashLoaderSWFO::$_swfo_atts

```
array = array(
        "xiredirecturl" => false,
        "redirecturl" => false,
        "detectkey" => false
        ) [line 83]
```

list of SWFObject attribute defaults

- **Link** http://blog.deconcept.com/swfobject/
- **Access** private

Constructor *void* function FlashLoaderSWFO::__construct($attr,...) *[line 116]*
Function Parameters:

- *string* **$attr,...** "attribute=value",... optional arguments

The purpose of the constructor is to initialize the main attribute list and then fill it with the arguments (if any).

- **Access** public

void function FlashLoaderSWFO::addMovie($attr,...) *[line 235]*

Function Parameters:

- *string* **$attr,...** "attribute=value",... optional arguments

Adds movie arguments to list of movies to be embedded later by loadMovies().

- **See** function loadMovies
- **Access** public

void function FlashLoaderSWFO::getCommonJS() *[line 217]*

Prints include to JavaScript library.

- **Access** public

void function FlashLoaderSWFO::getCommonJSPath() *[line 225]*

Prints include to JavaScript library.

- **Access** public

void function FlashLoaderSWFO::getMovie($attr,...) *[line 270]*

Function Parameters:

- *mixed* **$attr,...** "attribute=value",... optional arguments or array $atts list

Gets (embbeds) flash movie in (X)HTML document.

- **Access** public

void function FlashLoaderSWFO::getMovies() *[line 251]*

Gets movies in list created by addMovie()

- **See** function loadMovie
- **Access** public

boolean function FlashLoaderSWFO::hasMovies() *[line 243]*

Check to see if there are any movies to load.

- **Access** public

void function FlashLoaderSWFO::loadMovie($attr,...) *[line 297]*

Function Parameters:

- *mixed* **$attr,...** "attribute=value",... optional arguments or array $atts list

Loads (embbeds) flash movie in (X)HTML document.

- **Access** public

void function FlashLoaderSWFO::loadMovies() *[line 262]*

Loads movies in list created by addMovie()

- **See** function loadMovie
- **Access** public

void function FlashLoaderSWFO::setAtts($attr,...) *[line 126]*

Function Parameters:

- *string* **$attr,...** "attribute=value",... optional arguments

Fill the attribute list with the arguments (if any).

- **Access** public

string function FlashLoaderSWFO::_buildSWFObject($atts) *[line 185]*

Function Parameters:

- *array* **$atts**

Builds the SWFObject JavaScript calls.

- **Access** private

array function FlashLoaderSWFO::_getAtts($argv) *[line 138]*

Function Parameters:

- *array* **$argv** attribute/value string pairs

Parses attribute/value string pairs into an array and returns them combined with current attribute list.

- **Access** private

mixed function FlashLoaderSWFO::_missingReqAtt(&$atts, $atts) *[line 157]*

Function Parameters:

- *array* **$atts**
- **&$atts**

Checks attribute list against required attribute list.

- **Access** private

boolean function FlashLoaderSWFO::_optionalAtt($att) *[line 172]*

Function Parameters:

- *string* **$att** attribute name

Checks if attribute is optional.

- **Access** private

Class GDColor

[line 37]

The GDColor Class is used to easily handle colors while using the GD Library.

- **Package** dbdCommon

GDColor::$color

int = [line 56]

Integer color value to be used by GD functions.

- **Access** private

GDColor::$im

resource = [line 43]

GD Image resource to allocate colors for.

- **Access** private

GDColor::$rgba

array = [line 50]

Red, Green, Blue, and Alpha channel values stored in an array respectively.

- **Var** of ints
- **Access** private

Constructor *void* function GDColor::__construct($i, $hex) *[line 64]*

Function Parameters:

- *resource* **$i** of image from imagecreate...
- *string* **$hex** color value

The constructor's job is to convert HEX color to its integer parts and then allocate the color.

- **Access** public

int function GDColor::getAlpha() *[line 132]*

Get the integer color value for the Alpha channel.

- **Access** public

int function GDColor::getBlue() *[line 123]*

Get the integer color value for the Blue channel.

- **Access** public

int function GDColor::getColor() *[line 87]*

Get the integer color value for use in GD functions.

- **Access** public

int function GDColor::getGreen() *[line 114]*

Get the integer color value for the Green channel.

- **Access** public

string function GDColor::getHex() *[line 96]*

Get the HEX color value of the allocated color.

- **Access** public

int function GDColor::getRed() *[line 105]*

Get the integer color value for the Red channel.

- **Access** public

array function GDColor::hex2arr($hex) *[line 161]*

Function Parameters:

- *string* **$hex** color

Static function for breaking a HEX color into its various channels.

- **Static**
- **Access** public

array function GDColor::int2arr($int) *[line 144]*

Function Parameters:

- *int* **$int** color

Static function for breaking an integer color into its various channels.

- **Static**
- **Access** public

void function GDColor::__destruct() *[line 77]*

Clean up the allocated colors.

- **Access** public

Class InputCSS

[line 15]

The InputCSS Class is to generate the markup and JQuery calls to make a fancy CSS form input.

- **Package** dbdCommon

InputCSS::$div_id

int = 0 *[line 23]*

Static value to increment to keep track of divs.

- **Static**
- **Access** private

InputCSS::$input_ids

array = array() *[line 30]*

Static array of uses input ids.

- **Var** of ints
- **Static**
- **Access** private

string function InputCSS::button($id, $name, $value, [$disabled = false], [$extra = ""]) *[line 137]*

Function Parameters:

- *string* **$id**
- *string* **$name**
- *string* **$value**
- *bool* **$disabled**
- *string* **$extra**

Generate a submit button.

- **Static**
- **Access** public

string function InputCSS::checkbox($id, $name, $value, [$disabled = false], [$checked = false], [$extra = ""]) *[line 180]*

Function Parameters:

- *string* **$id**
- *string* **$name**
- *string* **$value**
- *bool* **$disabled**
- *bool* **$checked**
- *string* **$extra**

Generate a checkbox input.

- **Static**
- **Access** public

string function InputCSS::file($id, $name, [$disabled = false], [$extra = ""], $value) *[line 165]*

Function Parameters:

- *string* **$id**
- *string* **$name**
- *string* **$value**
- *bool* **$disabled**
- *string* **$extra**

Generate a file/browse input.

- **Static**
- **Access** public

string function InputCSS::getCallingFile() *[line 37]*

Get the file name of the calling file.

- **Static**
- **Access** private

string function InputCSS::getDivID() *[line 63]*

Generate id for div tag.

- **Static**
- **Access** private

string function InputCSS::getInputID($id) *[line 51]*

Function Parameters:

- *string* **$id**

Generate id for input tag.

- **Static**
- **Access** private

string function InputCSS::getXHTML($type, $id, $name, $value, $disabled, [$checked = false], [$extra = ""]) *[line 79]*

Function Parameters:

- *string* **$type**
- *string* **$id**
- *string* **$name**
- *string* **$value**
- *bool* **$disabled**

- *string* **$extra**
- **$checked**

Generate all markup required for final output.

- **Static**
- **Access** private

string function InputCSS::radio($id, $name, $value, [$disabled = false], [$checked = false], [$extra = ""]) *[line 195]*

Function Parameters:

- *string* **$id**
- *string* **$name**
- *string* **$value**
- *bool* **$disabled**
- *bool* **$checked**
- *string* **$extra**

Generate a radio input.

- **Static**
- **Access** public

string function InputCSS::reset($id, $name, $value, [$disabled = false], [$extra = ""]) *[line 151]*

Function Parameters:

- *string* **$id**
- *string* **$name**
- *string* **$value**
- *bool* **$disabled**
- *string* **$extra**

Generate a reset button.

- **Static**
- **Access** public

Class QTLoader

[line 29]

The QTLoader class is used to effortlessly embedd QuickTime movies in a (X)HTML Document.

REQUIRED ITEM: qtobject.js

Simple Usage:

```
$QT = new QTLoader();
$QT->  loadMovie("src=movies/movie.mov"          , "element_id=mainQuickTime"          ,
"width=800"          , "height=600"          );
```

Advanced Usage:

```
$QT = new QTLoader();
$QT->  addMovie("src=movies/movie1.mov"          , "element_id=movie1"          ,
"width=800"          , "height=600"          );
$QT->  addMovie("src=movies/movie2.mov"          , "element_id=movie2"          ,
"width=800"          , "height=600"          );
$QT->  loadMovies();
```

- **Package** dbdCommon

QTLoader::$atts

array = array(
"href" => false,
"target" => false,
"kioskmode" => false,
"bgcolor" => false,
"showlogo" => true,
"targetcache" => false,
"controller" => true,

"autoplay" => false
) *[line 84]*

list of main attribute defaults

- **Link** http://www.w3schools.com/media/media_quicktime.asp
- **Access** private

QTLoader::$class_id

string = "02BF25D5-8C17-4B23-BC80-D3488ABDDC6B" *[line 44]*

the quicktime plugin class id

- **Access** private

QTLoader::$codebase

string = "http://www.apple.com/qtactivex/qtplugin.cab" *[line 49]*

codebase without version number

- **Access** private

QTLoader::$common_js_shown

boolean = false *[line 59]*

flag to be set so we write JavaScript only once

- **Access** private

QTLoader::$js_path

string = "common_js/qtobject.js" *[line 39]*

path to SWFObject external JavaScripts library

- **Link** http://blog.deconcept.com/2005/01/26/web-standards-compliant-javascript-quicktime-detect-and-embed/
- **Access** private

QTLoader::$movies

array = array() *[line 66]*

list of movies added my addMovie()

- **See** function addMovie
- **Access** private

QTLoader::$onload

mixed = "" *[line 60]*

- **Access** private

QTLoader::$pluginspage

string = "http://www.apple.com/quicktime/download/" *[line 54]*

plugins page url

- **Access** private

QTLoader::$req_atts

array = array(
"src" => false,
"width" => false,
"height" => false,
"element_id" => false,
"id" => 1
) *[line 72]*

list of required attribute defaults

- **Link** http://www.w3schools.com/media/media_quicktime.asp
- **Access** private

Constructor *void* function QTLoader::__construct($attr,...) *[line 100]*
Function Parameters:

- *string* **$attr,...** "attribute=value",... optional arguments

The purpose of the constructor is to initialize the main attribute list and then fill it with the arguments (if any).

- **Access** public

void function QTLoader::addMovie($attr,...) *[line 219]*
Function Parameters:

- *string* **$attr,...** "attribute=value",... optional arguments

Adds movie arguments to list of movies to be embedded later by loadMovies().

- **See** function loadMovies
- **Access** public

string function QTLoader::buildQTObject($atts) *[line 176]*

Function Parameters:

- *array* **$atts**

Builds the QTObject JavaScript calls.

- **Access** private

array function QTLoader::getAtts($argv) *[line 122]*

Function Parameters:

- *array* **$argv** attribute/value string pairs

Parses attribute/value string pairs into an array and returns them combined with current attribute list.

- **Access** private

void function QTLoader::getCommonJS() *[line 201]*

Prints include to JavaScript library.

- **Access** public

void function QTLoader::getCommonJSPath() *[line 209]*

Prints include to JavaScript library.

- **Access** public

void function QTLoader::getMovie($attr,...) *[line 254]*

Function Parameters:

- *mixed* **$attr,...** "attribute=value",... optional arguments or array $atts list

Gets (embbeds) quicktime movie in (X)HTML document.

- **Access** public

void function QTLoader::getMovies() *[line 235]*

Gets movies in list created by addMovie()

- **See** function loadMovie
- **Access** public

boolean function QTLoader::hasMovies() *[line 227]*

Check to see if there are any movies to load.

- **Access** public

void function QTLoader::loadMovie($attr,...) *[line 281]*

Function Parameters:

- *mixed* **$attr,...** "attribute=value",... optional arguments or array $atts list

Loads (embbeds) quicktime movie in (X)HTML document.

- **Access** public

void function QTLoader::loadMovies() *[line 246]*

Loads movies in list created by addMovie()

- **See** function loadMovie
- **Access** public

mixed function QTLoader::missingReqAtt(&$atts, $atts) *[line 148]*

Function Parameters:

- *array* **$atts**
- **&$atts**

Checks attribute list against required attribute list.

- **Access** private

boolean function QTLoader::optionalAtt($att) *[line 163]*

Function Parameters:

- *string* **$att** attribute name

Checks if attribute is optional.

- **Access** private

void function QTLoader::setAtts($attr,...) *[line 110]*

Function Parameters:

- *string* **$attr,...** "attribute=value",... optional arguments

Fill the attribute list with the arguments (if any).

- **Access** public

Class wmArrays

[line 15]

The wmArrays class is used to preform complex routines on arrays.

- **Package** dbdCommon
- **Static**

array function wmArrays::arrayMergeRecursiveSumFields($arr1, $arr2, $fields) *[line 79]*

Function Parameters:

- *array* **$arr1** first array
- *array* **$arr2** second array
- *array* **$fields** list of string keys of values to sum

Merges multidimentional arrays, and sums the integer values of fields in given list.

Usage:

```
$arr1 = array(
        "page"          =>    "index"          ,
        "hits"          =>    10
        );
$arr2 = array(
        "page"          =>    "index"          ,
        "hits"          =>    20
        );
$marr = wmArrays::arrayMergeRecursiveSumFields($arr1, $arr2, array("hits"          ));
//result
$marr = array(
        "page"          =>    "index"          ,
        "hits"          =>    30
        );
```

- **Static**
- **Access** public

void function wmArrays::arrayRemoveValue($arr, $value) *[line 111]*

Function Parameters:

- **$arr**
- **$value**

- **Static**
- **Access** public

array function wmArrays::arraySearchRecursive($arr, $val, [$key = false], [$max_depth = -1]) *[line 30]*

Function Parameters:

- *array* **$arr** array to search through
- *mixed* **$val** value to search for in array
- *mixed* **$key** optional key to match value with
- *int* **$max_depth** optional maximum depth of recursion

Searches through a multidimensional array for a given value and if found, returns an array of keys to its location.

Usage:

```
$index = wmArrays::arraySearchRecursive($arr, "will@dontblinkdesign.com"         );
```

- **Static**
- **Access** public

void function wmArrays::arrayValuesRecursive($arr) *[line 96]*

Function Parameters:

- **$arr**

- **Static**
- **Access** public

void function wmArrays::explodeNewline($string) *[line 129]*

Function Parameters:

- **$string**

- **Static**
- **Access** public

void function wmArrays::implodeObjArray($glue, $arr, $method) *[line 134]*

Function Parameters:

- **$glue**
- **$arr**
- **$method**

- **Static**
- **Access** public

void function wmArrays::pregExplode($pattern, $string, [$limit = -1]) *[line 124]*

Function Parameters:

- **$pattern**
- **$string**
- **$limit**

- **Static**
- **Access** public

Class wmFileSystem

[line 17]

The wmFileSystem class is collection of specialized file system utility methods.

- **Package** dbdCommon
- **Static**

string function wmFileSystem::bytesToHuman($size) *[line 101]*

Function Parameters:

- *int* **$size** byte count

Makes byte counts human readble (i.e. 1024 -> 1KB).

Usage:

```
$size = wmFileSystem::bytesToHuman($size);
```

- **Static**
- **Access** public

integer function wmFileSystem::dirSize($dir, [$exclude = array()]) *[line 79]*

Function Parameters:

- *string* **$dir**
- **$exclude**

Get the file sixe of a directory and its subcontents in bytes

- **Static**
- **Access** public

string function wmFileSystem::mimeType($file) *[line 117]*

Function Parameters:

- *string* **$file**

Get the mime type from mime_magic

- **Static**
- **Access** public

array function wmFileSystem::scanDir($dir, [$max_depth = -1]) *[line 30]*

Function Parameters:

- *string* **$dir** target directory
- *int* **$max_depth** maximum depth for recursion

Reads directory contents recursively and returns them as an associative array.

Usage:

```
     $files = wmFileSystem::scanDir("./somedir/"         );
```

- **Static**
- **Access** public

Class wmSort

[line 15]

The wmSort class is used for advanced two-dimentional array sorting.

- **Package** dbdCommon
- **Static**

wmSort::$sortby

mixed = [line 17]

- **Static**
- **Access** private

void function wmSort::multiASort(&$arr, $sortby, $arr) *[line 43]*

Function Parameters:

- *array* **$arr** reference to associative array to be sorted
- *string* **$sortby** key(s) to sort by and direction
- **&$arr**

Sorts a two-dimentional array by one to many key "indexes" and maintains its associative indexes.

Usage:

```
wmSort::multiASort($arr, "name:desc,date:asc"          );
```

- **Static**
- **Access** public

void function wmSort::multiSort(&$arr, $sortby, $arr) *[line 27]*

Function Parameters:

- *array* **$arr** reference to array to be sorted
- *string* **$sortby** key(s) to sort by and direction
- **&$arr**

Sorts a two-dimentional array by one to many key "indexes".

Usage:

```
wmSort::multiSort($arr, "name:desc,date:asc"         );
```

- **Static**
- **Access** public

mixed function wmSort::_getVal(&$var, $field, $var) *[line 59]*

Function Parameters:

- *mixed* **$var** reference to variable containing a comapriable value
- *string* **$field** array index, or object method name to get value for comparison
- **&$var**

Finds value for comparison.

This method checks to see if the field given for sorting is an associative array, object, or string.

- **Static**
- **Access** private

int function wmSort::_wmMultiSort(&$var1, &$var2, [$sortby = false], $var1, $var2) *[line 92]*

Function Parameters:

- *mixed* **$var1** value of current array postion
- *mixed* **$var2** value of next array postion
- *string* **$sortby** optional sortby argument
- **&$var1**
- **&$var2**

Callback function passed to usort() and uasort().

This method does the actual value comparison.

- **Static**
- **Access** private

Class wmString

[line 15]

The wmString class is used to manipulate strings in various ways.

- **Package** dbdCommon
- **Static**

wmString::TIME_LEN_FORMAT_LONG

= 1 *[line 43]*

wmString::TIME_LEN_FORMAT_SHORT

= 0 *[line 42]*

wmString::TIME_LEN_LABEL_AGO

= "Ago" *[line 39]*

wmString::TIME_LEN_LABEL_DAY

= "Day" *[line 34]*

wmString::TIME_LEN_LABEL_HRS

= "Hour" *[line 33]*

wmString::TIME_LEN_LABEL_LAST

= "Last" *[line 38]*

wmString::TIME_LEN_LABEL_MIN

= "Minute" *[line 32]*

wmString::TIME_LEN_LABEL_MON

= "Month" *[line 36]*

wmString::TIME_LEN_LABEL_SEC

= "Second" *[line 31]*

wmString::TIME_LEN_LABEL_TODAY

= "Today" *[line 41]*

wmString::TIME_LEN_LABEL_WEEK

= "Week" *[line 35]*

wmString::TIME_LEN_LABEL_YEAR

= "Year" *[line 37]*

wmString::TIME_LEN_LABEL_YESTERDAY

= "Yesterday" *[line 40]*

wmString::TIME_LEN_TIME_DAY

= 86400 *[line 27]*

wmString::TIME_LEN_TIME_HRS

= 3600 *[line 26]*

wmString::TIME_LEN_TIME_MIN

= 60 *[line 25]*

wmString::TIME_LEN_TIME_MON

= 2592000 *[line 29]*

wmString::TIME_LEN_TIME_SEC

= 1 *[line 24]*

wmString::TIME_LEN_TIME_WEEK

= 604800 *[line 28]*

wmString::TIME_LEN_TIME_YEAR

= 31536000 *[line 30]*

wmString::TIME_LEN_TYPE_DAY

= 3 *[line 20]*

wmString::TIME_LEN_TYPE_HRS

= 2 *[line 19]*

wmString::TIME_LEN_TYPE_MIN

= 1 *[line 18]*

wmString::TIME_LEN_TYPE_MON

= 5 *[line 22]*

wmString::TIME_LEN_TYPE_SEC

= 0 *[line 17]*

wmString::TIME_LEN_TYPE_WEEK

= 4 *[line 21]*

wmString::TIME_LEN_TYPE_YEAR

= 6 *[line 23]*

wmString::VALID_DATE

= '/\\A(?:^((\\d{2}(([02468][048])|([13579][26]))[\\-\\/\\s]?((((0?[13578])|(1[02]))[\\-\\/\\s]?((0?[1-9])|([1-2][0-9])|(3[01])))|(((0?[469])|(11))[\\-\\/\\s]?((0?[1-9])|([1-2][0-9])|(30)))|(0?2[\\-\\/\\s]?((0?[1-9])|([1-2][0-9])))))|(\\d{2}(([02468][1235679])|([13579][01345789]))[\\-\\/\\s]?((((0?[13578])|(1[02]))[\\-\\/\\s]?((0?[1-9])|([1-2][0-9])|(3[01])))|(((0?[469])|(11))[\\-\\/\\s]?((0?[1-9])|([1-2][0-9])|(30)))|(0?2[\\-\\/\\s]?((0?[1-9])|(1[0-9])|(2[0-8]))))))(\\s(((0?[0-9])|(1[0-9])|(2[0-3]))\\:([0-5][0-9])((\\s)|(\\:([0-5][0-9])))?))?$)\\z/' *[line 45]*

wmString::VALID_EMAIL

= '/[a-z0-9!#$%&\'*+\/=?^_`{|}~-]+(?:\.[a-z0-9!#$%&\'*+\/=?^_`{|}~-]+)*@(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?/i' *[line 44]*

wmString::$number_suffixes

mixed = array('th', 'st', 'nd', 'rd') *[line 48]*

- **Static**
- **Access** private

wmString::$plural_exceptions

mixed = array('pain') *[line 49]*

- **Static**

- **Access** private

wmString::$random_word_cons

mixed = array('b', 'd', 'f', 'g', 'h', 'k', 'l', 'm', 'n', 'p', 'r', 's', 't', 'v', 'w', 'z', 'ng', 'ch', 'ty', 'ny', 'gy', 'py', 'by') *[line 46]*

- **Static**
- **Access** private

wmString::$random_word_vowl

mixed = array('a', 'e', 'i', 'o', 'u') *[line 47]*

- **Static**
- **Access** private

boolean function wmString::dateValid($date) *[line 369]*
Function Parameters:

- *string* **$date**

Test if a date is valid.

- **Static**
- **Access** public

string function wmString::dayOfWeek([$day_num = 0], [$format_string = "l"]) *[line 268]*
Function Parameters:

- *integer* **$day_num**
- *string* **$format_string**

Give the day of week given day number 0-6.

- **Static**
- **Access** public

string function wmString::emailEncode($str) *[line 60]*

Function Parameters:

- *string* **$str** email address and/or mailto

Takes a string (mailto:email@domain.com) and converts it to unicode to protect it from spiders.

Usage:

```
$email = wmString::emailEncode("mailto:email@domain.com"         );
```

- **Static**
- **Access** public

boolean function wmString::emailValid($email) *[line 359]*

Function Parameters:

- *string* **$email**

Test if an email address is valid.

- **Static**
- **Access** public

integer function wmString::hanukkahStart([$year = null]) *[line 284]*
Function Parameters:

- *integer* **$year**

Get the unix timestamp of Hanukkah for this year.

- **Static**
- **Access** public

string function wmString::jsSpecialChars($str) *[line 153]*
Function Parameters:

- *string* **$str**

Escapes special characters for use in javascript function calls.

- **Static**
- **Access** public

string function wmString::lcFirst($str) *[line 141]*
Function Parameters:

- *string* **$str**

Turns the first characher of a string to lower case

- **Static**
- **Access** public

string function wmString::makePlural($str) *[line 90]*
Function Parameters:

- *string* **$str**

Takes a string and converts to the plural version.

- **Static**
- **Access** public

string function wmString::makeSingular($str) *[line 74]*
Function Parameters:

- *string* **$str**

Takes a string and converts to the singular version.

- **Static**
- **Access** public

string function wmString::numberSuffix($num) *[line 296]*
Function Parameters:

- *mixed* **$num**

Add suffixes to numbers.

- **Static**
- **Access** public

string function wmString::randomWord([$length = 5]) *[line 317]*

Function Parameters:

- *integer* **$length**

Generate a random readable word.

- **Static**
- **Access** public

string function wmString::superCap2UnderLower($str) *[line 118]*

Function Parameters:

- *string* **$str**

Takes a super capped string (SuperCapString) and adds underscores and makes lowercase

- **Static**
- **Access** public

string function wmString::superCapAddSpaces($str) *[line 106]*

Function Parameters:

- *string* **$str**

Takes a super capped string (SuperCapString) and adds spaces

- **Static**
- **Access** public

string function wmString::timeFormat($seconds) *[line 247]*
Function Parameters:

- *mixed* **$seconds**

Format the amount of time in seconds to hh::mm:ss

- **Static**
- **Access** public

string function wmString::timeLengthFormat($time, [$type_in = self::TIME_LEN_TYPE_SEC], [$type_out = self::TIME_LEN_TYPE_SEC], [$format = self::TIME_LEN_FORMAT_LONG]) *[line 168]*
Function Parameters:

- *integer* **$time**
- *integer* **$type_in**
- *integer* **$type_out**
- *integer* **$format**

Format a length of time.

- **Static**
- **Access** public

string function wmString::timePastFormat($date, [$use_today = false]) *[line 211]*

Function Parameters:

- *mixed* **$date**
- **$use_today**

Format the abount of time past.

- **Static**
- **Access** public

string function wmString::truncate($str, [$len = 80], [$etc = "..."], [$break_words = false], [$middle = false]) *[line 338]*

Function Parameters:

- *string* **$str**
- *integer* **$len**
- *string* **$etc**
- *boolean* **$break_words**
- *boolean* **$middle**

Truncate a string...

- **Static**
- **Access** public

string function wmString::underLower2SuperCap($str) *[line 129]*

Function Parameters:

- *string* **$str**

Takes an underscored lower case string (under_lower_string) converts it to SuperCaps

- **Static**
- **Access** public

string function wmString::wkt2kml($wkt, [$name = false], [$desc = false], [$style_id = false]) *[line 382]*

Function Parameters:

- *string* **$wkt**
- *string* **$name**
- *string* **$desc**
- *string* **$style_id**

Convert WKT to KML.

- **Static**
- **Access** public

Package FirePHP Procedural Elements

FirePHP.class.php

***** BEGIN LICENSE BLOCK *******

***** END LICENSE BLOCK *****

- **Package** FirePHP
- **Author** Christoph Dorn < christoph@christophdorn.com>
- **Copyright** Copyright (C) 2007-2009 Christoph Dorn
- **License** http://www.opensource.org/licenses/bsd-license.php

Package FirePHP Classes

Class FirePHP

[line 58]

Sends the given data to the FirePHP Firefox Extension.

The data can be displayed in the Firebug Console or in the "Server" request tab.

For more information see: http://www.firephp.org/

- **Package** FirePHP
- **Author** Christoph Dorn < christoph@christophdorn.com>
- **Copyright** Copyright (C) 2007-2009 Christoph Dorn
- **License** http://www.opensource.org/licenses/bsd-license.php

FirePHP::DUMP

= 'DUMP' *[line 108]*

Dumps a variable to firebug's server panel

FirePHP::ERROR

= 'ERROR' *[line 101]*

Firebug ERROR level

Logs a message to firebug console, displays an error icon before the message and colors the line yellow. Also increments the firebug error count.

FirePHP::EXCEPTION

= 'EXCEPTION' *[line 124]*

Displays an exception in firebug console

Increments the firebug error count.

FirePHP::GROUP_END

= 'GROUP_END' *[line 145]*

Ends a group in firebug console

FirePHP::GROUP_START

= 'GROUP_START' *[line 138]*

Starts a group in firebug console

FirePHP::INFO

= 'INFO' *[line 83]*

Firebug INFO level

Logs a message to firebug console and displays an info icon before the message.

FirePHP::LOG

= 'LOG' *[line 74]*

Firebug LOG level

Logs a message to firebug console.

FirePHP::TABLE

= 'TABLE' *[line 131]*

Displays an table in firebug console

FirePHP::TRACE

= 'TRACE' *[line 115]*

Displays a stack trace in firebug console

FirePHP::VERSION

= '0.3' *[line 65]*

FirePHP version

FirePHP::WARN

= 'WARN' *[line 92]*

Firebug WARN level

Logs a message to firebug console, displays an warning icon before the message and colors the line turquoise.

FirePHP::$convertAssertionErrorsToExceptions

boolean = true *[line 173]*

Flag whether to convert PHP assertion errors to Exceptions

- **Access** protected

FirePHP::$enabled

boolean = true *[line 218]*

Flag to enable/disable logging

- **Access** protected

FirePHP::$inExceptionHandler

boolean = false *[line 159]*

Flag whether we are logging from within the exception handler

- **Access** protected

FirePHP::$instance

FirePHP = null *[line 152]*

Singleton instance of FirePHP

- **Static**
- **Access** protected

FirePHP::$json_objectStack

mixed = array() *[line 1236]*

Keep a list of objects as we descend into the array so we can detect recursion.

- **Access** private

FirePHP::$messageIndex

int = 1 *[line 187]*

Wildfire protocol message index

- **Access** protected

FirePHP::$objectFilters

array = array() *[line 204]*

Filters used to exclude object members when encoding

- **Access** protected

FirePHP::$objectStack

object = array() *[line 211]*

A stack of objects used to detect recursion during object encoding

- **Access** protected

FirePHP::$options

```
array = array('maxObjectDepth' => 10,
              'maxArrayDepth' => 20,
              'useNativeJsonEncode' => true,
              'includeLineNumbers' => true) [line 194]
```

Options for the library

- **Access** protected

FirePHP::$throwAssertionExceptions

boolean = false *[line 180]*

Flag whether to throw PHP assertion errors that have been converted to Exceptions

- **Access** protected

FirePHP::$throwErrorExceptions

boolean = true *[line 166]*

Flag whether to throw PHP errors that have been converted to ErrorExceptions

- **Access** protected

Constructor *void* function FirePHP::__construct() *[line 223]*

The object constructor

void function FirePHP::assertionHandler($file, $line, $code) *[line 421]*

Function Parameters:

- *string* **$file** File source of assertion
- *int* **$line** Line source of assertion
- *mixed* **$code** Assertion code

FirePHP's assertion handler

Logs all assertions to your firebug console and then stops the script.

- **Access** public

boolean function FirePHP::detectClientExtension() *[line 596]*

Check if FirePHP is installed on client

- **Access** public

true function FirePHP::dump($Key, $Variable) *[line 562]*

Function Parameters:

- *string* **$Key**
- *mixed* **$Variable**

Dumps key and variable to firebug server panel

- **See** FirePHP::DUMP
- **Throws** Exception
- **Access** public

array function FirePHP::encodeObject($Object, [$ObjectDepth = 1], [$ArrayDepth = 1], $Depth) *[line 1011]*

Function Parameters:

- *Object* **$Object** The object to be encoded
- *int* **$Depth** The current traversal depth
- **$ObjectDepth**
- **$ArrayDepth**

Encodes an object including members with protected and private visibility

- **Access** protected

array function FirePHP::encodeTable($Table) *[line 982]*

Function Parameters:

- *array* **$Table** The table to be encoded

Encodes a table by encoding each row and column with encodeObject()

- **Access** protected

true function FirePHP::error($Object, [$Label = null]) *[line 549]*
Function Parameters:

- *mixes* **$Object**
- *string* **$Label**

Log object with label to firebug console

- **See** FirePHP::ERROR
- **Throws** Exception
- **Access** public

void function FirePHP::errorHandler($errno, $errstr, $errfile, $errline, $errcontext) *[line 344]*
Function Parameters:

- *int* **$errno**
- *string* **$errstr**
- *string* **$errfile**
- *int* **$errline**
- *array* **$errcontext**

FirePHP's error handler

Throws exception for each php error that will occur.

- **Access** public

void function FirePHP::exceptionHandler($Exception) *[line 382]*

Function Parameters:

- *Exception* **$Exception**

FirePHP's exception handler

Logs all exceptions to your firebug console and then stops the script.

- **Throws** Exception

true function FirePHP::fb($Object) *[line 613]*

Function Parameters:

- *mixed* **$Object** The variable to be logged

Log varible to Firebug

- **See** http://www.firephp.org/Wiki/Reference/Fb
- **Throws** Exception
- **Access** public

boolean function FirePHP::getEnabled() *[line 272]*

Check if logging is enabled

- **Access** public

FirePHP function FirePHP::getInstance([$AutoCreate = false]) *[line 241]*

Function Parameters:

- *boolean* **$AutoCreate**

Gets singleton instance of FirePHP

- **Static**
- **Access** public

array function FirePHP::getOptions() *[line 310]*

Get options from the library

- **Access** public

string|*false* function FirePHP::getUserAgent() *[line 938]*

Get user agent

- **Access** protected

true function FirePHP::group($Name, [$Options = null]) *[line 473]*

Function Parameters:

- *string* **$Name**
- *array* **$Options** OPTIONAL Instructions on how to log the group

Start a group for following messages.

Options: Collapsed: [true|false] Color: [#RRGGBB|ColorName]

- **Throws** Exception
- **Access** public

true function FirePHP::groupEnd() *[line 497]*

Ends a group you have started before

- **Throws** Exception
- **Access** public

true function FirePHP::info($Object, [$Label = null]) *[line 523]*

Function Parameters:

- *mixes* **$Object**
- *string* **$Label**

Log object with label to firebug console

- **See** FirePHP::INFO
- **Throws** Exception
- **Access** public

FirePHP function FirePHP::init() *[line 253]*

Creates FirePHP object and stores it for singleton access

- **Static**

- **Access** public

boolean function FirePHP::is_utf8($str) *[line 1150]*
Function Parameters:

- *mixed* **$str** String to be tested

Returns true if $string is valid UTF-8 and false otherwise.

- **Static**
- **Access** protected

string function FirePHP::jsonEncode($Object, [$skipObjectEncode = false]) *[line 961]*
Function Parameters:

- *object* **$Object** The object to be encoded
- **$skipObjectEncode**

Encode an object into a JSON string
Uses PHP's jeson_encode() if available

- **Access** public

mixed function FirePHP::json_encode($var) *[line 1294]*
Function Parameters:

- *mixed* **$var** any number, boolean, string, array, or object to be encoded. see argument 1 to Services_JSON() above for array-parsing behavior. if var is a strng, note that encode() always expects it to be in ASCII or UTF-8 format!

encodes an arbitrary variable into JSON format

- **Access** private

string function FirePHP::json_name_value($name, $value) *[line 1509]*
Function Parameters:

- *string* **$name** name of key to use
- *mixed* **$value** reference to an array element to be encoded

array-walking function for use in generating JSON-formatted name-value pairs

- **Access** private

string function FirePHP::json_utf82utf16($utf8) *[line 1250]*
Function Parameters:

- *string* **$utf8** UTF-8 character

convert a string from one UTF-8 char to one UTF-16 char

Normally should be handled by mb_convert_encoding, but provides a slower PHP-only method for installations that lack the multibye string extension.

- **Access** private

true function FirePHP::log($Object, [$Label = null]) *[line 510]*

Function Parameters:

- *mixes* **$Object**
- *string* **$Label**

Log object with label to firebug console

- **See** FirePHP::LOG
- **Throws** Exception
- **Access** public

Exception function FirePHP::newException($Message) *[line 949]*

Function Parameters:

- *string* **$Message**

Returns a new exception

- **Access** protected

mixed function FirePHP::registerAssertionHandler([$convertAssertionErrorsToExceptions = true], [$throwAssertionExceptions = false]) *[line 400]*

Function Parameters:

- *boolean* **$convertAssertionErrorsToExceptions**
- *boolean* **$throwAssertionExceptions**

Register FirePHP driver as your assert callback

- **Access** public

mixed function FirePHP::registerErrorHandler([$throwErrorExceptions = true]) *[line 321]*

Function Parameters:

- **$throwErrorExceptions**

Register FirePHP as your error handler

Will throw exceptions for each php error.

- **Access** public

mixed function FirePHP::registerExceptionHandler() *[line 369]*

Register FirePHP as your exception handler

- **Access** public

void function FirePHP::setEnabled($Enabled) *[line 263]*

Function Parameters:

- *boolean* **$Enabled** TRUE to enable, FALSE to disable

Enable and disable logging to Firebug

- **Access** public

void function FirePHP::setHeader($Name, $Value) *[line 929]*

Function Parameters:

- *string* **$Name**
- *string_type* **$Value**

Send header

- **Access** protected

void function FirePHP::setObjectFilter($Class, $Filter) *[line 285]*

Function Parameters:

- *string* **$Class** The class name of the object
- *array* **$Filter** An array of members to exclude

Specify a filter to be used when encoding an object

Filters are used to exclude object members.

- **Access** public

void function FirePHP::setOptions($Options) *[line 301]*

Function Parameters:

- *array* **$Options** The options to be set

Set some options for the library

Options:

- maxObjectDepth: The maximum depth to traverse objects (default: 10)
- maxArrayDepth: The maximum depth to traverse arrays (default: 20)
- useNativeJsonEncode: If true will use json_encode() (default: true)
- includeLineNumbers: If true will include line numbers and filenames (default: true)

- **Access** public

void function FirePHP::setProcessorUrl($URL) *[line 446]*

Function Parameters:

- *string* **$URL**

Set custom processor url for FirePHP

- **Access** public

void function FirePHP::setRendererUrl($URL) *[line 456]*

Function Parameters:

- *string* **$URL**

Set custom renderer url for FirePHP

- **Access** public

true function FirePHP::table($Label, $Table) *[line 587]*

Function Parameters:

- *string* **$Label**
- *string* **$Table**

Log a table in the firebug console

- **See** FirePHP::TABLE
- **Throws** Exception
- **Access** public

true function FirePHP::trace($Label) *[line 574]*

Function Parameters:

- *string* **$Label**

Log a trace in the firebug console

- **See** FirePHP::TRACE
- **Throws** Exception
- **Access** public

true function FirePHP::warn($Object, [$Label = null]) *[line 536]*

Function Parameters:

- *mixes* **$Object**
- *string* **$Label**

Log object with label to firebug console

- **See** FirePHP::WARN
- **Throws** Exception
- **Access** public

array function FirePHP::_escapeTrace($Trace) *[line 892]*
Function Parameters:

- *array* **$Trace**

Escape trace path for windows systems

- **Access** protected

string function FirePHP::_escapeTraceFile($File) *[line 911]*
Function Parameters:

- *string* **$File**

Escape file information of trace for windows systems

- **Access** protected

string function FirePHP::_standardizePath($Path) *[line 882]*
Function Parameters:

- *string* **$Path**

Standardizes path for windows systems.

- **Access** protected

array function FirePHP::__sleep() *[line 231]*

When the object gets serialized only include specific object members.

- **Access** public

Package JSMin Classes

Class JSMin

[line 48]

jsmin.php - PHP implementation of Douglas Crockford's JSMin.

This is pretty much a direct port of jsmin.c to PHP with just a few PHP-specific performance tweaks. Also, whereas jsmin.c reads from stdin and outputs to stdout, this library accepts a string as input and returns another string as output.

PHP 5 or higher is required.

Permission is hereby granted to use this version of the library under the same terms as jsmin.c, which has the following license:

-- Copyright (c) 2002 Douglas Crockford (www.crockford.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. --

- **Package** JSMin
- **Author** Ryan Grove < ryan@wonko.com>
- **Version** 1.1.0 (2007-06-01)
- **Copyright** 2002 Douglas Crockford <douglas@crockford.com> (jsmin.c)
- **Copyright** 2007 Ryan Grove <ryan@wonko.com> (PHP port)
- **Link** http://code.google.com/p/jsmin-php/
- **License** MIT License

JSMin::ORD_LF

= 10 *[line 49]*

JSMin::ORD_SPACE

= 32 *[line 50]*

JSMin::$a

mixed = '' *[line 52]*

- **Access** protected

JSMin::$b

mixed = '' *[line 53]*

- **Access** protected

JSMin::$input

mixed = '' *[line 54]*

- **Access** protected

JSMin::$inputIndex

mixed = 0 *[line 55]*

- **Access** protected

JSMin::$inputLength

mixed = 0 *[line 56]*

- **Access** protected

JSMin::$lookAhead

mixed = null *[line 57]*

- **Access** protected

JSMin::$output

mixed = array() *[line 58]*

- **Access** protected

Constructor *void* function JSMin::__construct($input) *[line 69]*

Function Parameters:

- **$input**

- **Access** public

void function JSMin::action($d) *[line 76]*

Function Parameters:

- **$d**

- **Access** protected

void function JSMin::get() *[line 138]*

- **Access** protected

void function JSMin::isAlphaNum($c) *[line 163]*

Function Parameters:

- **$c**

- **Access** protected

void function JSMin::min() *[line 167]*

- **Access** protected

void function JSMin::minify($js) *[line 62]*

Function Parameters:

- **$js**

- **Static**
- **Access** public

void function JSMin::next() *[line 249]*

- **Access** protected

void function JSMin::peek() *[line 288]*

- **Access** protected

Class JSMinException

[line 295]

- **Package** JSMin

Package dbdMVC Procedural Elements

dbdMVC.php

dbdMVC.php :: dbdMVC Include File & Front Controller Class

dbdMVC version 1.8.2 Copyright (c) 2006-2009 by Don't Blink Design http://dbdmvc.com

dbdMVC is a light weight model-view-controller framework.

dbdMVC is released under the terms of the LGPL license http://www.gnu.org/copyleft/lesser.html

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.8.2
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design
- **TODO** Create readme and update dbdMVC description
- **License** http://www.gnu.org/copyleft/lesser.html

void function dbdLog([$msg = ""], [$type = null], [$dest = null]) *[line 90]*

Function Parameters:

- **$msg**
- **$type**

- **$dest**

DBD_CACHE_DIR = DBD_APP_DIR."cache".DBD_DS *[line 68]*

dbdMVC cache directory

DBD_DEBUG_ALL = 255 *[line 76]*

Turn on all debugging

DBD_DEBUG_CSS = 8 *[line 83]*
DBD_DEBUG_DB = 1 *[line 80]*

Turn on all debugging

DBD_DEBUG_HTML = 2 *[line 81]*
DBD_DEBUG_JS = 4 *[line 82]*
DBD_DS = DIRECTORY_SEPARATOR *[line 39]*

Shorthand for DIRECTORY_SEPARATOR

DBD_FIREPHP_DIR = DBD_MVC_DIR."FirePHPCore-0.3.1".DBD_DS."lib".DBD_DS *[line 60]*

FirePHP core directory

DBD_MVC_CLI = php_sapi_name()=="cli" *[line 72]*

Are we running in CLI mode?

DBD_PLUG_DIR = DBD_MVC_DIR."plugins".DBD_DS *[line 64]*

dbdMVC plug-in directory

DBD_SMARTY_DIR = DBD_MVC_DIR."Smarty-2.6.18".DBD_DS."libs".DBD_DS *[line 48]*

Smarty package directory

DBD_SMARTY_PLUG_DIR = DBD_SMARTY_DIR.DBD_DS."plugins_dbd".DBD_DS *[line 52]*

Smarty add0n plug-in directory directory (used to avoid disrupting standard plug-in dir)

DBD_XAJAX_DIR = DBD_MVC_DIR."xajax_0.2.5".DBD_DS *[line 56]*

xAjax package directory

require_once **"lib/dbdLoader.php"** *[line 111]*

After the dbdLoader is included, all further dbdMVC includes use dbdLoader::load() or __autoload()

require_once **FirePHP.class.php** *[line 89]*

error_log function that handles complex types too

fb.php

- **Package** dbdMVC

true function fb([$Object =]) *[line 57]*
Function Parameters:

- *mixed* **$Object**

Sends the given data to the FirePHP Firefox Extension.
The data can be displayed in the Firebug Console or in the "Server" request tab.

- **See** http://www.firephp.org/Wiki/Reference/Fb
- **Throws** Exception

require_once **dirname(__FILE__).'/FirePHP.class.php'** *[line 45]*

dbconn.inc

dbconn.inc :: The Database string constants

- **Package** dbdMVC
- **Author** Will Mason < will@dontblinkdesign.com>
- **Version** 1.2
- **Copyright** Copyright (c) 2009 by Don't Blink Design

DBCONN_DEFAULT_DB = '' *[line 18]*

Constant default database name if none is specified at connection time

DBCONN_INCLUDED = true *[line 13]*

Constant boolean to flag file inclusion.

DB_DRIVER = 'mysql' *[line 23]*

Database connection strings

DB_HOST = 'localhost' *[line 24]*
DB_PASS = 'i$th3$hit' *[line 27]*
DB_PORT = '3306' *[line 25]*
DB_USER = 'dbdmvc' *[line 26]*

Index.php

Index.php :: Index Controller Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.1
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

index.php

index.php :: dbdMVC Bootstrap Sample File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.4
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

DBD_APP_DIR = realpath(DBD_DOC_ROOT."/../dbdApp")."/" *[line 11]*
DBD_DOC_ROOT = dirname(__FILE__) *[line 10]*
DBD_MVC_DIR = "/var/www/dbdMVC/" *[line 12]*
require_once **DBD_MVC_DIR."dbdMVC.php"** *[line 15]*

dbdController.php

dbdController.php :: dbdController Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.9
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

dbdCSS.php

dbdCSS.php :: dbdCSS Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.12
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

dbdDB.php

dbdDB.php :: dbdDB Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.3
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

dbdDispatcher.php

dbdDispatcher.php :: dbdDispatcher Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.8
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

dbdEmptyController.php

dbdEmptyController.php :: dbdEmptyController Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.1
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

dbdError.php

dbdError.php :: dbdError Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.2
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

dbdException.php

dbdException.php :: dbdException Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.3
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

dbdHoldableException.php

dbdHoldableException.php :: dbdHoldableException Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.1
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

dbdInfo.php

dbdInfo.php :: dbdInfo Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.1
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

dbdJS.php

dbdJS.php :: dbdJS Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.7
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

dbdLoader.php

dbdLoader.php :: dbdLoader Class File & __autoload() Function

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.6
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

require_once **"dbdException.php"** *[line 14]*

dbdException class

void function __autoload($class) *[line 237]*

Function Parameters:

- *string* **$class**

Autoload classes using dbdLoader::loadClass();

dbdModel.php

dbdModel.php :: dbdModel Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.11
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

dbdRequest.php

dbdRequest.php :: dbdRequest Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.2
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

dbdRouter.php

dbdRouter.php :: dbdRouter Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.3
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

dbdSmarty.php

dbdSmarty.php :: dbdSmarty Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.14
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

dbdURI.php

dbdURI.php :: dbdURI Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.8
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

dbdXajax.php

dbdXajax.php :: dbdXajax Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.1
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

dbdXajaxJS.php

dbdXajaxJS.php :: dbdXajaxJS Class File

- **Package** dbdMVC
- **Author** Don't Blink Design < info@dontblinkdesign.com>
- **Version** 1.2
- **Copyright** Copyright (c) 2006-2009 by Don't Blink Design

wmGD.php

- **Package** dbdMVC

void function colorizeImage(&$im, $color) *[line 100]*

Function Parameters:

- **&$im**
- *GDColor* **$color**

void function cropImage(&$src, $w, $h) *[line 168]*

Function Parameters:

- **&$src**
- **$w**
- **$h**

void function cropImageReal(&$src, $x, $y, $w, $h) *[line 239]*

Function Parameters:

- **&$src**
- **$x**
- **$y**
- **$w**
- **$h**

void function cropResizeImage(&$src, $w, $h, [$align = 1]) *[line 192]*

Function Parameters:

- **&$src**
- **$w**
- **$h**
- **$align**

void function imageCreateFrom($filename) *[line 4]*

Function Parameters:

- **$filename**

void function imagefilledellipsealpha(&$im, $cx, $cy, $w, $h, $color) *[line 281]*
Function Parameters:

- **&$im**
- **$cx**
- **$cy**
- **$w**
- **$h**
- **$color**

void function imageLineThick(&$im, $x1, $y1, $x2, $y2, $color, [$thick = 1]) *[line 260]*
Function Parameters:

- **&$im**
- **$x1**
- **$y1**
- **$x2**
- **$y2**
- **$color**
- **$thick**

void function imageType($filename) *[line 76]*
Function Parameters:

- **$filename**

void function imageTypesArray() *[line 82]*
void function outputAndDestroy(&$im, [$type = "png"], [$cache_file = null], [$no_output = false]) *[line 20]*
Function Parameters:

- **&$im**
- **$type**
- **$cache_file**
- **$no_output**

void function reflectImage(&$src, [$ref_height = 0.50], [$fade_start = 80], [$fade_end = 0], [$tint = "7f7f7f"], [$bg = "ffffff"], [$type = "png"]) *[line 355]*
Function Parameters:

- **&$src**
- **$ref_height**
- **$fade_start**
- **$fade_end**
- **$tint**
- **$bg**
- **$type**

require_once **"GDColor.php"** *[line 2]*

void function resizeImage(&$src, $w, $h, [$force_upscale = false]) *[line 142]*

Function Parameters:

- **&$src**
- **$w**
- **$h**
- **$force_upscale**

void function stretchCopyButton($dst_im, $src_im, $dst_wd, $src_wd, $src_ht, [$y = 0]) *[line 248]*

Function Parameters:

- **$dst_im**
- **$src_im**
- **$dst_wd**
- **$src_wd**
- **$src_ht**
- **$y**

Package dbdMVC Classes

Class dbdController

[line 22]

Abstract Parent Controller Class All controller classes must extend this class or a child of this class.

- **Package** dbdMVC
- **TODO** Use Response object for output.
- **Abstract Element**
- **Usedby** dbdInfo
- **Usedby** dbdJS
- **Usedby** dbdDispatcher
- **Usedby** Index
- **Usedby** dbdError
- **Usedby** dbdCSS
- **Uses** dbdLoader
- **Uses** dbdSmarty
- **Uses** dbdXajax
- **Uses** dbdController::$xajaxResponse
- **Usedby** dbdEmptyController

dbdController::$app_dir

string = null *[line 32]*

Application directory

- **Access** private

dbdController::$default_render

boolean = true *[line 43]*

Should dbdController attempt to render
a template on destruction?

- **Access** private

dbdController::$model

object = null *[line 59]*

Instance of some model.
Note: May not be used.

- **Access** protected

dbdController::$router

dbdRouter = null *[line 53]*

Instance of dbdRouter

- **Access** protected

dbdController::$template

string = null *[line 37]*

Template filename

- **Access** private

dbdController::$view

dbdSmarty = null *[line 64]*

Instance of dbdSmarty

- **Access** protected

dbdController::$xajax

dbdXajax = null *[line 69]*

Instance of dbdXajax

- **Access** protected

dbdController::$xajaxResponse

dbdXajaxResponse = null *[line 75]*

Instance of xajaxResponse object

- **TODO** make dbdXajaxResponse object
- **Access** protected
- **Usedby** dbdController

Constructor *void* function dbdController::__construct($router, $app_dir) *[line 87]*

Function Parameters:

- *dbdRouter* **$router**
- *string* **$app_dir**

Prepare controller for action.

Set the router and app_dir, create the view, and then call init(). **Note:** The constructor is declared final to prevent corruption of object. Any further construction needed by child classes can be accomplished by overriding dbdController::init().

- **Final**
- **Access** public

void function dbdController::assignAllParams() *[line 239]*

Assign all request parameters to the view as an associative array.

Note: Slashes are stripped.

- **Access** protected

void function dbdController::autoExec() *[line 144]*

Auto execute method to be executed after the action.

Can be overridden for additional automatic functionality (maybe assign some variable to the view). **Note:** If overriden, parent::autoExec(); must be called to maintain current functionality!

- **Access** public

void function dbdController::autoRender() *[line 126]*

Auto render the view.

Can be disabled by using $this->noRender(); If a template has not been set, it will attempt to convert the controller name into a template filename.

- **Final**
- **Throws** dbdException
- **Access** public

void function dbdController::dnit() *[line 171]*

Default dnit method.

This method is to be overridden for additional object destruction and de-initialization (maybe cleaning up the $model property).

- **Access** protected

void function dbdController::doDefault() *[line 177]*

Default action method.

Can be preformed without the presense of a child controller.

- **Access** public

void function dbdController::doXajax() *[line 184]*

Prepare controller for xajax request.

Ajax actions a called by dbdXajax.

- **Final**
- **Access** public

void function dbdController::forward([$url = "/"]) *[line 248]*

Function Parameters:

- *string* **$url**

Preform an HTTP Location change.

Disables auto render.

- **Access** protected

string function dbdController::getAction() *[line 273]*

Get action name.

- **Access** protected

string function dbdController::getAppDir() *[line 257]*

Get application directory.

- **Access** protected

string function dbdController::getController() *[line 265]*

Get controller name.

- **Access** protected

mixed function dbdController::getParam($name) *[line 299]*

Function Parameters:

- *string* **$name**

Get request parameter.

- **Access** protected

array function dbdController::getParams() *[line 308]*

Get all parameters.

- **See** dbdRouter::getParams()
- **Access** protected

string function dbdController::getTemplate() *[line 231]*

Get template filename.

- **Access** protected

array function dbdController::getUploadInfo($id) *[line 330]*

Function Parameters:

- *string* **$id**

Get the progress information on a current upload via APC.

- **Link** http://us.php.net/manual/en/apc.configuration.php#ini.apc.rfc1867
- **Access** protected
- **Uses** php-pecl-apc

string function dbdController::getURL([$get_params = false], [$host = false]) *[line 319]*

Function Parameters:

- *boolean* **$get_params**
- *boolean* **$host**

Get current request url.

If optional flag is passed, parameters are included.

- **Access** protected

void function dbdController::init() *[line 162]*

Default init method.

This method is to be overridden for additional object construction and initialization (maybe setting the $model property).

- **Access** protected

void function dbdController::noRender() *[line 151]*

Disable auto render on destruction.

- **Access** public

void function dbdController::registerXajaxActions([$controller = null]) *[line 204]*

Function Parameters:

- *string* **$controller**

Register controller actions with dbdXajax.

If controller name is not passed it defaults to the current one. JavaScript is assigned to view as $xajax. **Note:** Acion methods must use the dbdDispatcher::XAJAX_ACTION_PREFIX.

- **TODO** Auto insert JS into header of view output using some filter.
- **Final**
- **Access** protected

void function dbdController::setParam($name, $value) *[line 282]*

Function Parameters:

- *string* **$name**
- *mixed* **$value** value

Set request parameter.

- **Access** protected

void function dbdController::setTemplate($tpl) *[line 223]*

Function Parameters:

- *string* **$tpl**

Set template filename.

- **Access** protected

void function dbdController::unsetParam($name) *[line 290]*

Function Parameters:

- *string* **$name**

Unset request parameter.

- **Access** protected

void function dbdController::__destruct($router, $app_dir) *[line 114]*

Function Parameters:

- *dbdRouter* **$router**
- *string* **$app_dir**

Conclude the controller action.

Call dnit(). **Note:** The destructor is declared final to prevent corruption of object. Any further destruction needed by child classes can be accomplished by overriding dbdController::dnit().

- **Final**
- **Access** public

Class dbdCSS

[line 63]

Controller class for compressing and serving css files.

Can be used to combine files using @import statments and the doCombine action. Also extends CSS by adding internal/external variable support and simple math. **Set Variable Example:**

- **Package** dbdMVC
- **Uses** dbdException
- **Uses** dbdController

dbdCSS::BLOCK_CLOSE_REGEX

= '/[}]/' *[line 85]*

Css block close pattern.

dbdCSS::BLOCK_OPEN_REGEX

= '/[{]/' *[line 81]*

Css block open pattern.

dbdCSS::BUTT_DEF_TAG_REGEX

= '/\b([#.][a-z][a-z0-9#.-]+)?(a|input)([#.])([a-z][a-z0-9#.-]*)[,]?/i' *[line 122]*

Css button definition pattern.

dbdCSS::BUTT_GEN_TAG_REGEX

= '/[@]generated/i' *[line 114]*

Css generated buttons at-tag pattern.

dbdCSS::BUTT_GEN_TPL_TAG_REGEX

= '/[#]([a-z][a-z0-9]*)/i' *[line 118]*

Css generated buttons template tag pattern.

dbdCSS::BUTT_PROP_DEF_DISABLED_REGEX

= '/[-]disabled$/i' *[line 134]*

Css button property disabled pattern.

dbdCSS::BUTT_PROP_DEF_HOVER_REGEX

= '/[-]hover$/i' *[line 130]*

Css button property hover pattern.

dbdCSS::BUTT_PROP_DEF_REGEX

= '/([a-z][a-z-]*)[\t\n]*[:][\t\n]*([^;]+)[;]/i' *[line 126]*

Css button property definition pattern.

dbdCSS::BUTT_TAG_REGEX

= '/^[@]buttons/i' *[line 110]*

Css buttons at-tag pattern.

dbdCSS::CACHE_INFO_FILES

= '/^[ ]?*[ ]?[@]files (.+)$/i' *[line 142]*

Css caching info file list pattern.

dbdCSS::CACHE_INFO_SPRITE

= '/^[ ]?*[ ]?[@]sprite (.+)$/i' *[line 138]*

Css caching info sprite pattern.

dbdCSS::CSS_EXT_REGEX

= '/^.+\.[c][s][s]$/i' *[line 77]*

Css file extension pattern.

dbdCSS::DEBUG_REGEX

= '/^\/*.*[@]debug.**\//i' *[line 73]*

Debug comment to turn off minification.

dbdCSS::DIR_DELIM_REGEX

= "/[,\|]/" *[line 146]*

Directory delimiter for passing a string of files

dbdCSS::IMPORT_REGEX

= '/^[@]import[ ][\'\"]([a-z0-9-_\.]+[\.][c][s][s])[\'\"][;]/i' *[line 69]*

Css import regular expression.

- **TODO** allow for url() syntax

dbdCSS::MATH_CALL_REGEX

= '/math[(](.+)[)]/ie' *[line 102]*

Css math call pattern. (for simple equations)

dbdCSS::MATH_SAFE_REGEX

= '!^[0-9()%/+*-]+$!i' *[line 106]*

Css math safe characters pattern. (for simple equations)

dbdCSS::VAR_CALL_REGEX

= '/var[(]([a-z][a-z0-9-_]*)[)]/ie' *[line 98]*

Css variable call pattern. (for dereferencing)

dbdCSS::VAR_DEF_REGEX

= '/([a-z][a-z0-9-_]*)[\t\n]*[:][\t\n]*([^;]+)[;]/i' *[line 94]*

Css variable definition pattern.

dbdCSS::VAR_TAG_REGEX

= '/^[@]variables/i' *[line 90]*

Css variables at-tag pattern.

Uses implimentation specs from http://disruptive-innovations.com/zoo/cssvariables/

dbdCSS::$buffer

string = "" *[line 311]*

Output buffer that will be minified if debug is off.

- **Access** private

dbdCSS::$buttons

array = array() *[line 266]*

List of css buttons that have been defined.

- **Access** private

dbdCSS::$buttons_generated

array = array() *[line 261]*

List of css buttons that have been generated.

- **Access** private

dbdCSS::$button_props

```
array = array(
      'global' => array(
         'offset' => 0,
         'transparent' => '#7fffffff',
         'format' => 'png',
         'cache' => null,
         'src' => null
      ),'generated_global'=>array(),'generated_tpl'=>array('font'=>null,'font-size'=>null,'font-color'=>null,'min-
text-width'=>null,//        'text-align' => 'center',
'text-decoration'=>'none','text-vertical-offset'=>0,'cap-width'=>null,'src-body'=>null,'src-shade'=>null,'src-
high'=>null),'generated_def'=>array('generated'=>null,'text'=>null,'background-color'=>'#7fffffff','foreground-
color'=>null,'font-color'=>null,'src-body'=>null,'src-shade'=>null,'src-high'=>null),'def'=>array('background-
image'=>null)) [line 171]
```

List of available css button properties.

- **Access** private

dbdCSS::$button_props_generated_global

array = array() *[line 251]*

List of css generated button global properties.

- **Access** private

dbdCSS::$button_props_generated_tpl

array = array() *[line 256]*

List of css generated button template properties.

- **Access** private

dbdCSS::$button_props_global

array = array() *[line 246]*

List of css button global properties.

- **Access** private

dbdCSS::$button_props_required

```
array = array(
      'global' => array(
         'offset',
         'transparent',
         'format',
         'cache',
         'src'
      ),'generated_global'=>array(),'generated_tpl'=>array('font','font-size','font-color','min-text-width','cap-
width'),'generated_def'=>array('generated','text','background-color','foreground-color','font-color',//         'text-
align',
'text-decoration','text-vertical-offset','src-body'),'def'=>array('background-image')) [line 211]
```

List of required css button properties.

- **Access** private

dbdCSS::$cache_dir

string = null *[line 276]*

Cache directory

- **Access** private

dbdCSS::$cache_file

string = null *[line 281]*

Cache file name

- **Access** private

dbdCSS::$debug

boolean = false *[line 271]*

Flag to turn off minification.

- **Access** private

dbdCSS::$file

string = "" *[line 296]*

Current parser file

- **Access** private

dbdCSS::$files

array = array() *[line 286]*

List of files for proccessing

- **Var** string
- **Access** private

dbdCSS::$line

string = "" *[line 301]*

Current parser line

- **Access** private

dbdCSS::$line_num

integer = 0 *[line 306]*

Current parser line number

- **Access** private

dbdCSS::$minify_regex

array = array(
    "spaces" => array("/ [ ]+/", "
"),"whitespace"=>array("/(\n|\t)+/",""),"spec_char_left"=>array("/([{}:;]+)\s+/","\\1"),"spec_char_right"=>array("/\s
+([{}:;]+)/","\\1"),"comments"=>array("%(?(?=/*[^/]*?\*/)(/*.*?\\*/[^/]*?/*.*?*/)|/*.*?*/)%","\\1")) *[line 155]*

Array of regex pairs for minifying the css.

- **Access** private

dbdCSS::$sprite

string = null *[line 291]*

Sprite file name

- **Access** private

dbdCSS::$vars

array = array() *[line 166]*

List of css varables that have been defined.

- **Access** private

boolean function dbdCSS::checkButtonCache($type, $params) *[line 961]*
Function Parameters:

- *string* **$type**
- *array* **$params**

Check for a cached button.

- **Access** private

string function dbdCSS::checkButtonPropsRequired($key, [$selector = null]) *[line 983]*

Function Parameters:

- *string* **$key**
- *string* **$selector**

Check that the required property exists.

- **Access** private

boolean function dbdCSS::checkCache($files) *[line 326]*

Function Parameters:

- **$files**

Check for a cache file and make sure its not outdated.

- **Access** private

void function dbdCSS::createCache() *[line 374]*

Create cache file and dump buffer.

- **Access** private

void function dbdCSS::doCombine() *[line 1168]*

Serve multiple css files as one

- **Access** public

void function dbdCSS::dumpFile(&$fp, $fp) *[line 1087]*

Function Parameters:

- *mixed* **$fp**
- **&$fp**

Dump and close a file.

Note: Can except a string file name or open resource.

- **Access** private

void function dbdCSS::ensureResource(&$fp, $fp) *[line 1121]*

Function Parameters:

- *mixed* **$fp**
- **&$fp**

Ensure a resource is open and valid.

- **Throws** dbdException
- **Access** private

string function dbdCSS::genButton($id, $p) *[line 852]*

Function Parameters:

- *integer* **$id**
- *array* **$p**

Generate a layered button from source images.

- **Access** private

string function dbdCSS::genButtonCacheFile($type, $params) *[line 943]*
Function Parameters:

- *string* **$type**
- *array* **$params**

Generate the cached sprite file name.

- **Access** private

void function dbdCSS::genButtonCSS() *[line 678]*
Generate css for the defined buttons.

- **Access** private

void function dbdCSS::genButtons() *[line 834]*
Genrate layered buttons from source images.

- **Access** private

string function dbdCSS::genButtonSprite() *[line 751]*

Generate the sprite with all of the buttons and register their offsets in the properties array.

- **Access** private

string function dbdCSS::genCacheName($files) *[line 316]*

Function Parameters:

- **$files**

Generate a cache file name based on the list of proccess files.

- **Access** private

string function dbdCSS::genURL([$files = array()], [$vars = array()]) *[line 1200]*

Function Parameters:

- *array* **$files**
- *array* **$vars**

Generate a combine url from an array of files

- **Static**
- **Access** public

array function dbdCSS::getButtonProps($key, [$selector = null]) *[line 999]*

Function Parameters:

- *string* **$key**
- *string* **$selector**

Get properties for a button

- **Access** private

void function dbdCSS::getVar($name) *[line 1067]*

Function Parameters:

- **$name**

- **Access** private

void function dbdCSS::init() *[line 1141]*

Set debug

- **Access** public

void function dbdCSS::math($eq) *[line 1075]*

Function Parameters:

- **$eq**

- **Access** private

void function dbdCSS::minify() *[line 1110]*

Minify buffer.

- **Access** private

void function dbdCSS::output([$cache = false]) *[line 1097]*

Function Parameters:

- **$cache**

Set headers, minify, and echo buffer.

- **Access** private

void function dbdCSS::parseImports($file, $dir) *[line 394]*

Function Parameters:

- *string* **$file**
- *string* **$dir**

Parse css files for import statements and recurse when found.

Dump files upon no match.

- **Throws** dbdException
- **Access** private

void function dbdCSS::setHeaders() *[line 1131]*

Set css headers

- **Access** private

Class dbdDB

[line 15]

dbdMVC Database Abstraction Layer

- **Package** dbdMVC

dbdDB::COMP_BETWEEN

= 12 *[line 45]*

dbdDB::COMP_EQ

= 1 *[line 34]*

dbdDB::COMP_GT

= 5 *[line 38]*

dbdDB::COMP_GTEQ

= 4 *[line 37]*

dbdDB::COMP_IN

= 8 *[line 41]*

dbdDB::COMP_LIKE

= 10 *[line 43]*

dbdDB::COMP_LT

= 7 *[line 40]*

dbdDB::COMP_LTEQ

= 6 *[line 39]*

dbdDB::COMP_NBETWEEN

= 13 *[line 46]*

dbdDB::COMP_NEQ

= 2 *[line 35]*

dbdDB::COMP_NIN

= 9 *[line 42]*

dbdDB::COMP_NLIKE

= 11 *[line 44]*

dbdDB::COMP_NULLEQ

= 3 *[line 36]*

dbdDB::COMP_TYPE

= "dbdDB::COMP_TYPE" *[line 33]*

dbdDB::DATE_FORMAT

= "Y-m-d H:i:s" *[line 26]*

SQL compliant date formating string

- **Var** constant

dbdDB::DBCONN_PATH

= "constant/dbconn.inc" *[line 21]*

path to database connection string file

- **Var** constant

dbdDB::TIME_FORMAT

= "H:i:s" *[line 31]*

SQL compliant time formating string

- **Var** constant

dbdDB::$instances

array = array() *[line 53]*

array of dbdDB objects

to be filled by the static getInstance method

- **Var** dbdDB
- **Static**
- **Access** private

dbdDB::$options

```
array = array(
      PDO::ATTR_CASE => PDO::CASE_LOWER,
      PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION
   ) [line 63]
```

Array of PDO options to set after construction

- **Access** private

dbdDB::$pdo

PDO = null *[line 58]*

PDO instance variable

- **Access** private

Constructor *void* function dbdDB::__construct($dsn, [$user = ""], [$pass = ""], [$driver_options = array()]) *[line 75]*

Function Parameters:

- *string* **$dsn**
- *string* **$user**
- *string* **$pass**
- *array* **$driver_options**

The constructors job is to instantiate the needed PDO object and set attributes

- **Access** private

string function dbdDB::assembleDSN([$db = ""]) *[line 131]*

Function Parameters:

- *string* **$db**

Assemble driver specific DSN's

- **Throws** dbdException
- **Static**
- **Access** private

string function dbdDB::date([$timestamp = null]) *[line 146]*

Function Parameters:

- *integer* **$timestamp**

Format date for SQL insert

- **Static**
- **Access** public

dbdDB function dbdDB::getInstance([$dsn = ""], [$user = ""], [$pass = ""], [$driver_options = array()]) *[line 98]*

Function Parameters:

- *string* **$dsn**
- *string* **$user**
- *string* **$pass**
- *array* **$driver_options**

Include db_conn string file.

Key method of Singleton Pattern!!

- **Static**
- **Access** public

void function dbdDB::includeDBConn() *[line 119]*

Include dbconn string file.

- **Throws** dbdException
- **Static**
- **Access** private

int function dbdDB::nextAutoID($table) *[line 178]*

Function Parameters:

- **$table**

Selects the next auto_increment id

- **Access** public

PDOStatement function dbdDB::prepExec($statement, [$input_parameters = array()], [$driver_options = array()]) *[line 167]*

Function Parameters:

- *string* **$statement**
- *array* **$input_parameters**
- *array* **$driver_options**

Prepare & excute a query statement

- **Throws** dbdException
- **Access** public

void function dbdDB::setAttributes() *[line 83]*

Set default PDO attributes

- **Access** private

string function dbdDB::time([$timestamp = null]) *[line 155]*

Function Parameters:

- *integer* **$timestamp**

Format time for SQL insert

- **Static**
- **Access** public

mixed function dbdDB::__call($name, $args) *[line 192]*

Function Parameters:

- *string* **$name**
- *array* **$args**

Magic function to call PDO functions and rethrow any exceptions as dbdException

- **Throws** dbdException
- **Access** public

Class dbdDispatcher

[line 19]

Dispatch controller actions.

- **Package** dbdMVC
- **Usedby** dbdMVC
- **Usedby** dbdXajax
- **Uses** dbdRouter
- **Uses** dbdLoader
- **Uses** dbdEmptyController
- **Uses** dbdController

dbdDispatcher::ACTION_PREFIX

= "do" *[line 48]*

Action method name prefix

dbdDispatcher::BUILTIN_PREFIX

= "dbd" *[line 24]*

Builtin controller class prefix

dbdDispatcher::DEFAULT_ACTION

= "default" *[line 40]*

Default action name

dbdDispatcher::DEFAULT_CONTROLLER

= "index" *[line 36]*

Default controller name

dbdDispatcher::ERROR_CONTROLLER

= "dbdError" *[line 32]*

Error controller class if is set

dbdDispatcher::FALLBACK_CONTROLLER

= "dbdEmptyController" *[line 28]*

Fallback controller class if none exists

dbdDispatcher::MAGIC_ACTION

= "action" *[line 44]*

Magic action name

dbdDispatcher::XAJAX_ACTION_PREFIX

= "xDo" *[line 52]*

Xajax action method name prefix

dbdDispatcher::$app_dir

string = null *[line 66]*

Application directory

- **Access** private

dbdDispatcher::$controller_dirs

string = array() *[line 71]*

Controller directory

- **Access** private

dbdDispatcher::$fallback_controller

string = null *[line 76]*

User defined fallback controller

- **Access** private

dbdDispatcher::$router

object = null *[line 61]*

Instance of dbdRouter

- **Access** private

Constructor *void* function dbdDispatcher::__construct([$router = null], [$app_dir = null]) *[line 83]*

Function Parameters:

- *dbdRouter* **$router**
- *string* **$app_dir**

Set router, app_dir, and controller_dir.

- **Access** public

void function dbdDispatcher::dispatch() *[line 95]*

Dispatch actions

- **Access** public

string function dbdDispatcher::getAction($controller) *[line 152]*

Function Parameters:

- *dbdController* **$controller**

Get action method name.

Trys controller name from router, or default action.

- **Throws** dbdException
- **Access** private

object function dbdDispatcher::getController() *[line 112]*

Get controller object.

Trys controller name from router, then fallback controller.

- **Throws** dbdException
- **Access** private

Class dbdEmptyController

[line 16]

Empty controller class for rendering templates without a controller!

- **Package** dbdMVC
- **Usedby** dbdDispatcher
- **Uses** dbdController

Class dbdError

[line 16]

Error handling controller class.

- **Package** dbdMVC
- **Uses** dbdController

dbdError::ERR_400

= "Bad Request" *[line 21]*

HTTP Status 400 Description

dbdError::ERR_401

= "Unauthorized" *[line 25]*

HTTP Status 401 Description

dbdError::ERR_403

= "Forbidden" *[line 29]*

HTTP Status 403 Description

dbdError::ERR_404

= "Not Found" *[line 33]*

HTTP Status 404 Description

dbdError::ERR_408

= "Request Timeout" *[line 37]*

HTTP Status 408 Description

dbdError::ERR_500

= "Internal Server Error" *[line 41]*

HTTP Status 500 Description

void function dbdError::doDefault() *[line 45]*

Attept to render error.tpl or just echo error.

- **Access** public

void function dbdError::doError($that) *[line 54]*

Function Parameters:

- *dbdController* **$that**

Crazy static action method expects intstance of dbdController for dbdError extension.

- **Static**
- **Access** public

Class dbdException

[line 15]

dbdMVC Exception class.

- **Package** dbdMVC
- **Usedby** dbdLoader
- **Usedby** dbdXajaxJS
- **Usedby** dbdJS
- **Usedby** dbdCSS

Constructor *void* function dbdException::__construct([$message = null], [$code = 500]) *[line 22]*

Function Parameters:

- *string* **$message**
- *integer* **$code**

Pass paremeters on to parent exception with prefixed message.

- **Access** public

Class dbdHoldableException

[line 15]

dbdMVC Exception class.

- **Package** dbdMVC

dbdHoldableException::$held

mixed = array() *[line 26]*

Array of held exceptions

- **Static Variable Used** array 0: dbdHoldableException
- **Static**
- **Access** protected

dbdHoldableException::$hold

mixed = false *[line 21]*

Flag holding status

- **Static Variable Used** boolean 0:
- **Static**
- **Access** protected

array function dbdHoldableException::getHeld() *[line 73]*

Get array of held exceptions

- **Static**
- **Access** public

void function dbdHoldableException::hold() *[line 31]*

Enable holding

- **Static**
- **Access** public

void function dbdHoldableException::intercept($e) *[line 46]*

Function Parameters:

- *dbdHoldableException* **$e**

Exception handler and method to call when ready to throw

- **Throws** dbdHoldableException
- **Static**
- **Access** public

void function dbdHoldableException::release() *[line 58]*

Disable holding and throw the first held excetion, if any

- **Throws** dbdHoldableException
- **Static**
- **Access** public

Class dbdInfo

[line 18]

Controller for displaying phpInfo.

Use /dbdInfo/.

- **Package** dbdMVC
- **Uses** dbdMVC
- **Uses** dbdController

void function dbdInfo::doDefault() *[line 23]*

Render phpinfo() if PHPisExposed or forward home.

- **Access** public

Class dbdJS

[line 18]

Controller class for compressing and serving js files.

Can be used to combine files using the doCombine action. Files are combined, minified, and cached once for later use.

- **Package** dbdMVC
- **Usedby** dbdXajaxJS
- **Uses** dbdException
- **Uses** dbdController

dbdJS::DIR_DELIM_REGEX

= "/[,\|]/" *[line 23]*

Directory delimiter for passing a string of files

dbdJS::$buffer

string = "" *[line 58]*

Output buffer that will be minified if debug is off.

- **Access** private

dbdJS::$cache_dir

string = null *[line 37]*

Cache directory

- **Access** private

dbdJS::$cache_file

string = null *[line 42]*

Cache file name

- **Access** private

dbdJS::$debug

boolean = false *[line 32]*

Flag to turn off minification.

- **Access** private

dbdJS::$files

array = array() *[line 47]*

List of files for proccessing

- **Var** string
- **Access** private

dbdJS::$vars

array = array() *[line 53]*

List of external varables to be added as properties of the dbdJS javascript object.

- **Access** private

void function dbdJS::addFile($file, $dir) *[line 163]*
Function Parameters:

- *string* **$file**
- *string* **$dir**

Check a file for existence and then add it to the array for later proccessing.

- **Access** protected

void function dbdJS::addVars() *[line 117]*
Add external variables to the buffer.

- **Access** private

boolean function dbdJS::checkCache() *[line 74]*

Check for a cache file and make sure its not outdated.

- **Access** private

void function dbdJS::createCache() *[line 94]*

Create cache file and dump buffer.

- **Access** private

void function dbdJS::doCombine() *[line 218]*

Serve multiple js files as one

- **Access** public

void function dbdJS::doDefault() *[line 203]*

Alias of doGet()

- **Access** public

void function dbdJS::doGet() *[line 210]*

Serve js files...

- **Access** public

void function dbdJS::ensureResource(&$fp, $fp) *[line 107]*
Function Parameters:

- *mixed* **$fp**
- **&$fp**

Ensure a resource is open and valid.

- **Throws** dbdException
- **Access** private

string function dbdJS::genCacheName() *[line 63]*

Generate a cache file name based on the list of proccess files.

- **Access** private

string function dbdJS::genURL([$files = array()], [$vars = array()]) *[line 238]*
Function Parameters:

- *array* **$files**
- *array* **$vars**

Generate a combine url from an array of files

- **Static**
- **Access** public

void function dbdJS::init() *[line 191]*

Set no render, cache_dir, debug, and load JSMin

- **Access** protected

void function dbdJS::minify() *[line 142]*

Minify buffer.

- **Access** private

void function dbdJS::output() *[line 175]*

Set headers, minify, and echo buffer.

- **Access** protected

void function dbdJS::readFiles($fp) *[line 128]*

Function Parameters:

- *mixed* **$fp**

Read and close a file.

Note: Can except a string file name or open resource.

- **Access** private

void function dbdJS::setHeaders() *[line 150]*

Set js headers

- **Access** private

Class dbdLoader

[line 20]

dbdMVC File and Class Loader

- **Package** dbdMVC
- **Usedby** dbdMVC
- **Usedby** dbdDispatcher
- **Usedby** dbdController
- **Uses** dbdException

dbdLoader::DEFAULT_CONTROLLER_DIR

= "controllers" *[line 25]*

Controller directory

dbdLoader::DEFAULT_MODEL_DIR

= "models" *[line 29]*

Model directory

dbdLoader::$controller_dirs

array = array() *[line 48]*

Controller directory

- **Access** private

dbdLoader::$include_paths

array = array() *[line 43]*

Array of directories to find classes and other files

- **Var** strings
- **Access** private

dbdLoader::$instance

mixed = null *[line 38]*

Active instance of dbdLoader Singleton

- **Static Variable Used** object 0:
- **Static**
- **Access** private

dbdLoader::$model_dir

string = null *[line 53]*

Model directory

- **Access** private

Constructor *void* function dbdLoader::__construct() *[line 59]*

Initialize the loader Note: **The cunstructer is private because this class can only be instantiated by dbdLoader::getInstance() to achieve a Singleton.**

- **Access** private

void function dbdLoader::addControllerDir($dir) *[line 135]*

Function Parameters:

- *string* **$dir**

Add a controller directory

- **Static**
- **Access** public

void function dbdLoader::addPath($path) *[line 111]*

Function Parameters:

- *string* **$path**

Add path to dbdLoader and php.ini

- **Static**
- **Access** public

string function dbdLoader::getControllerDirs() *[line 151]*

Get controller directory

- **Static**
- **Access** public

object instance function dbdLoader::getInstance() *[line 101]*

Get the active instance or create one.

- **Static**
- **Access** public

string function dbdLoader::getModelDir() *[line 159]*

Get model directory

- **Static**
- **Access** public

array function dbdLoader::getPaths() *[line 119]*

Get all paths registered with dbdLoader

- **Static**

- **Access** public

void function dbdLoader::init() *[line 66]*

Register php.ini include_path(s), controller dir, and model dir

- **Access** private

boolean function dbdLoader::load($file, [$dir = null]) *[line 193]*

Function Parameters:

- *string* **$file**
- *string* **$dir**

Load a php file using require_once if not already loaded.

- **Throws** dbdException
- **Static**
- **Access** public

void function dbdLoader::loadClass($class) *[line 211]*

Function Parameters:

- *string* **$class**

Load a class file. Calls dbdLoader::load().

Called by __autoload()

- **Static**
- **Access** public

void function dbdLoader::register($path) *[line 79]*

Function Parameters:

- *string* **$path**

Add path to array and to php.ini

- **Access** private

mixed function dbdLoader::search($file, [$dir = null]) *[line 170]*

Function Parameters:

- *string* **$file**
- *string* **$dir**

Search for a file in the list of path(s).

Returns path to file if found and false upon failure.

- **Static**
- **Access** public

void function dbdLoader::setControllerDir($dir) *[line 127]*

Function Parameters:

- *string* **$dir**

Set controller directory

- **Static**
- **Access** public

void function dbdLoader::setModelDir($dir) *[line 143]*

Function Parameters:

- *string* **$dir**

Set model directory

- **Static**
- **Access** public

Class dbdModel

[line 35]

dbdMVC Database Table Model

Simple Model Example

```
class Widget extends dbdModel
{
    const TABLE_NAME = "widgets"          ;
    const TABLE_KEY = "widget_id"          ;

    public function __construct($id = 0)
    {
        parent::__construct(__CLASS__, $id);
    }

    public static function getAll()
    {
        return parent::getAll(__CLASS__);
    }
}
```

- **Package** dbdMVC
- **Abstract Element**

dbdModel::CONST_TABLE_KEY

= 'TABLE_KEY' *[line 38]*

dbdModel::CONST_TABLE_NAME

= 'TABLE_NAME' *[line 37]*

dbdModel::$data

array = array() *[line 68]*

Array of fields from row

- **Access** protected

dbdModel::$db

dbdDB = null *[line 48]*

Static instance of dbdDB

- **Static**
- **Access** protected

dbdModel::$id

integer = 0 *[line 63]*

Row id

- **Access** protected

dbdModel::$reflections

array = array() *[line 43]*

A list of class reflections to limit overhead

- **Var** ReflectionClass
- **Static**
- **Access** private

dbdModel::$table_key

string = null *[line 58]*

The key for the table

- **Access** private

dbdModel::$table_name

string = null *[line 53]*

The name of the table

- **Access** private

Constructor *void* function dbdModel::__construct($class, [$id = 0]) *[line 75]*

Function Parameters:

- *string* **$class**
- *integer* **$id**

The constructer's job is to set the table info, get the db, and call the initializer.

- **Access** public

string function dbdModel::buildWhereClause([&$table_keys = array()], $table_keys) *[line 194]*
Function Parameters:

- *array* **$table_keys**
- **&$table_keys**

Build where clause for count and getAll methods

- **Static**
- **Access** private

void function dbdModel::delete() *[line 147]*
Delete this row

- **Access** public

void function dbdModel::ensureDB() *[line 296]*
Make sure we have an instance of the db

- **Static**
- **Access** public

array function dbdModel::getAll($class, [$table_keys = array()], [$order = null], [$limit = null], [$ids_only = false]) *[line 175]*

Function Parameters:

- *string* **$order**
- *string* **$limit**
- *boolean* **$ids_only**
- **$class**
- **$table_keys**

Get all rows from this table

- **Static**
- **Access** public

mixed function dbdModel::getConstant($class, $constant) *[line 287]*

Function Parameters:

- *string* **$class**
- *string* **$constant**

Return the value of a class constant using Reflection

- **Static**
- **Access** private

array function dbdModel::getCount($class, [$table_keys = array()]) *[line 160]*

Function Parameters:

- *string* **$class**
- *array* **$table_keys**

Get a count of all rows from this table

- **Static**
- **Access** public

array function dbdModel::getData() *[line 330]*

Get the fields for this row

- **Access** public

dbdDB function dbdModel::getDB() *[line 305]*

Get the database object

- **Static**
- **Access** public

integer function dbdModel::getID() *[line 322]*

Get the row id

- **Access** public

ReflectionClass function dbdModel::getReflection($class) *[line 275]*

Function Parameters:

- *string* **$class**

Return the Reflection of a class

- **Static**
- **Access** private

void function dbdModel::init() *[line 89]*

Select all the fields for this row

- **Access** protected

void function dbdModel::initFields() *[line 97]*

Select all the fields names for this table

- **Access** protected

void function dbdModel::save([$fields = array()]) *[line 110]*

Function Parameters:

- *array* **$fields**

Save all the fields for this row

- **Access** public

dbdDB function dbdModel::setDB($db) *[line 314]*
Function Parameters:

- *dbdDB* **$db**

Set the database object

- **Static**
- **Access** public

mixed function dbdModel::__call($name, $args) *[line 377]*
Function Parameters:

- *string* **$name**
- *array* **$args**

Magic function allows calling of extra magic functions for testing field values.
Inlcudes: has, is, get, and set.

- **Throws** dbdException
- **Access** public

mixed function dbdModel::__get($name) *[line 348]*
Function Parameters:

- *string* **$name**

Magic function for getting field values

- **Access** public

boolean function dbdModel::__isset($name) *[line 357]*
Function Parameters:

- *string* **$name**

Magic function to check if a field is set

- **Access** public

void function dbdModel::__set($name, $value) *[line 339]*
Function Parameters:

- *string* **$name**
- *mixed* **$value**

Magic function for setting field values

- **Access** public

void function dbdModel::__unset($name) *[line 365]*

Function Parameters:

- *string* **$name**

Magic function to unset a field

- **Access** public

Class dbdMVC

[line 129]

dbdMVC Front Controller Class

- **Package** dbdMVC
- **Usedby** dbdInfo
- **Usedby** dbdSmarty
- **Uses** dbdRouter
- **Uses** dbdRequest
- **Uses** dbdLoader
- **Uses** dbdDispatcher

dbdMVC::VERSION

= "1.8.2" *[line 135]*

Current version number

Note: May not match file @version number

dbdMVC::$app_dir

string = null *[line 169]*

Path to dbdApp files (models, views, and controllers dirs)

- **Access** private

dbdMVC::$app_name

string = null *[line 164]*

Name of dbdApp

- **Access** private

dbdMVC::$debug_mode

integer = 0 *[line 159]*

Current debug mode.

- **Access** private

dbdMVC::$dispatcher

object = null *[line 199]*

Instance of dbdDispatcher

- **Access** private

dbdMVC::$error_controller

string = null *[line 179]*

Error/Default controller (if left null, the dbdDispatcher::ERROR_CONTROLLER will be used)

- **Access** private

dbdMVC::$error_log

string = null *[line 184]*

Error log file (if left null, system default will be used)

- **Access** private

dbdMVC::$expose_php_info

mixed = null *[line 154]*

Allow phpinfo() via dbdInfo controller

- **Static Variable Used** boolean 0:
- **Static**
- **Access** private

dbdMVC::$fallback_controller

string = null *[line 174]*

Fallback/Default controller (if left null, the dbdDispatcher::FALLBACK_CONTROLLER will be used)

- **Access** private

dbdMVC::$instance

mixed = null *[line 144]*

Active instance of dbdMVC Singleton

- **Static Variable Used** object 0:
- **Static**
- **Access** private

dbdMVC::$request

object = null *[line 189]*

Instance of dbdRequest

- **Access** private

dbdMVC::$router

object = null *[line 194]*

Instance of dbdRouter

- **Access** private

dbdMVC::$start

mixed = null *[line 149]*

Start time of dbdApp, from microtime()

- **Static Variable Used** float 0:
- **Static**
- **Access** private

Constructor *void* function dbdMVC::__construct() *[line 205]*

Constructer sets exception handler and the default app_name.

Note: The cunstructer is private because this class can only be instantiated by dbdMVC::getInstance() to achieve a Singleton.

- **Access** private

boolean function dbdMVC::debugMode($mode) *[line 465]*

Function Parameters:

- **$mode**

Check if debug mode is set.

- **Static**
- **Access** public

void function dbdMVC::dispatch() *[line 232]*

Dispatch dbdDispatcher.

- **Access** private

void function dbdMVC::e($e) *[line 345]*

Function Parameters:

- *Exception* **$e**

Default ecpetion handler

- **Static**
- **Access** public

void function dbdMVC::exposePHPInfo() *[line 475]*

Allow phpinfo() display via dbdInfo.

WARNING: Allowing system information to be view by the public can be a major security risk. This should be used with caution for development only!

- **See** dbdInfo
- **Static**
- **Access** public

void function dbdMVC::genCLIURI() *[line 240]*

Generate a URI from the Command Line Arguments.

Usage: php index.php -c controller -a action --param=value

- **Access** private

string function dbdMVC::getAppDir() *[line 399]*

Get application directory.

- **Static**
- **Access** public

string function dbdMVC::getAppName() *[line 391]*

Get application name.

- **Static**
- **Access** public

string function dbdMVC::getErrorController() *[line 415]*

Get error controller name.

- **Static**
- **Access** public

float function dbdMVC::getErrorLog() *[line 423]*

Return the error_log file name.

- **Static**
- **Access** public

float function dbdMVC::getExecutionTime() *[line 447]*

Calculate execution time of the application and return.

- **Static**
- **Access** public

string function dbdMVC::getFallbackController() *[line 407]*

Get fallback controller name.

- **Static**
- **Access** public

object instance function dbdMVC::getInstance() *[line 335]*

Get the active instance of dbdMVC or create one.

- **Static**
- **Access** public

object dbdRequest function dbdMVC::getRequest() *[line 431]*

Get the instance of dbdRequest.

- **Static**
- **Access** public

object dbdRouter function dbdMVC::getRouter() *[line 439]*

Get the instance of dbdRouter.

- **Static**
- **Access** public

void function dbdMVC::init() *[line 215]*

Prepare for dispatch by setting up the dbdDispatcher and by adding the application specific paths to dbdLoader.

- **Access** private

void function dbdMVC::logExecutionTime() *[line 455]*

Log execution time of the application to the error_log.

- **Throws** dbdException
- **Static**
- **Access** public

boolean function dbdMVC::PHPisExposed() *[line 483]*

Check if phpinfo display is allowed.

- **Static**
- **Access** public

void function dbdMVC::run($app_dir, [$app_name = null], [$fallback_controller = null], [$error_controller = null], [$error_log = null], [$debug_mode = null]) *[line 369]*

Function Parameters:

- *string* **$app_dir**
- *string* **$app_name**

- **$fallback_controller**
- **$error_controller**
- **$error_log**
- **$debug_mode**

Run dbdMVC on the given dbdApp dir.

This is the main method of dbdMVC.

- **Static**
- **Access** public

void function dbdMVC::setAppDir($app_dir) *[line 317]*

Function Parameters:

- *string* **$app_dir**

Set application directory property.

Note: Accepts paths with or without trailing slash.

- **Static**
- **Access** public

void function dbdMVC::setAppName($app_name) *[line 284]*

Function Parameters:

- *string* **$app_name**

Set application name property.

- **Static**
- **Access** public

void function dbdMVC::setDebugMode($mode) *[line 327]*

Function Parameters:

- *integer* **$mode**

Set debug mode.

- **Static**
- **Access** public

void function dbdMVC::setErrorController($error) *[line 300]*

Function Parameters:

- *string* **$error**

Set error_controller property.

- **Static**
- **Access** public

void function dbdMVC::setErrorLog($path, $error) *[line 308]*

Function Parameters:

- *string* **$error**
- **$path**

Set error_log property.

- **Static**
- **Access** public

void function dbdMVC::setFallbackController($fallback) *[line 292]*

Function Parameters:

- *string* **$fallback**

Set fallback_controller property.

- **Static**
- **Access** public

Class dbdRequest

[line 16]

HTTP Request Class.

Based on Zend Framework Request Class

- **Package** dbdMVC
- **Usedby** dbdMVC
- **Usedby** dbdRouter

dbdRequest::$params

array = array() *[line 28]*

Additional parameters set at runtime since the $_SUPER_GLOBAL's seem to be read only.

- **Access** private

dbdRequest::$request_uri

string = null *[line 23]*

HTTP request URI

- **Access** private

Constructor *void* function dbdRequest::__construct([$uri = null]) *[line 34]*
Function Parameters:

- *string* **$uri**

Prepare the request object for use by setting the request uri.
If a string uri is not provided, it is pulled from $_SERVER['REQUEST_URI'].

- **Access** public

mixed function dbdRequest::get($key) *[line 80]*
Function Parameters:

- *string* **$key**

Access values contained in the superglobals as public members

Order of precedence: 1. GET, 2. POST, 3. COOKIE, 4. SERVER, 5. ENV

- **See** http://msdn.microsoft.com/en-us/library/system.web.httprequest.item.aspx
- **Access** public

mixed function dbdRequest::getCookie([$key = null], [$default = null]) *[line 176]*

Function Parameters:

- *string* **$key**
- *mixed* **$default** Default value to use if key not found

Retrieve a member of the $_COOKIE superglobal.

If no $key is passed, returns the entire $_COOKIE array.

- **TODO** How to retrieve from nested arrays
- **Access** public

mixed function dbdRequest::getEnv([$key = null], [$default = null]) *[line 202]*

Function Parameters:

- *string* **$key**
- *mixed* **$default** Default value to use if key not found

Retrieve a member of the $_ENV superglobal.

If no $key is passed, returns the entire $_COOKIE array.

- **Access** public

string|*false* function dbdRequest::getHeader($header) *[line 216]*

Function Parameters:

- *string* **$header** HTTP header name

Return the value of the given HTTP header. Pass the header name as the plain, HTTP-specified header name. Ex.: Ask for 'Accept' to get the Accept header, 'Accept-Encoding' to get the Accept-Encoding header.

- **Throws** dbdException
- **Access** public

mixed function dbdRequest::getPost([$key = null], [$default = null]) *[line 162]*

Function Parameters:

- *string* **$key**
- *mixed* **$default** Default value to use if key not found

Retrieve a member of the $_POST superglobal.

If no $key is passed, returns the entire $_POST array.

- **TODO** How to retrieve from nested arrays
- **Access** public

mixed function dbdRequest::getQuery([$key = null], [$default = null]) *[line 148]*

Function Parameters:

- *string* **$key**
- *mixed* **$default** Default value to use if key not found

Retrieve a member of the $_GET superglobal.

If no $key is passed, returns the entire $_GET array.

- **TODO** How to retrieve from nested arrays
- **Access** public

void function dbdRequest::getRequestURI($uri) *[line 60]*

Function Parameters:

- *string* **$uri**

Get request uri

- **Access** public

mixed function dbdRequest::getServer([$key = null], [$default = null]) *[line 189]*

Function Parameters:

- *string* **$key**
- *mixed* **$default** Default value to use if key not found

Retrieve a member of the $_SERVER superglobal.

If no $key is passed, returns the entire $_COOKIE array.

- **Access** public

boolean function dbdRequest::has($key) *[line 135]*

Function Parameters:

- *string* **$key**

Alias to __isset()

- **Access** public

boolean function dbdRequest::isXmlHttpRequest() *[line 241]*

Is the request a Javascript XMLHttpRequest? Should work with Prototype/Script.aculo.us, possibly others.

- **Access** public

void function dbdRequest::set($key, $val) *[line 69]*

Function Parameters:

- *string* **$key**
- *mixed* **$val**

Set parameters at runtime

- **Access** public

void function dbdRequest::setRequestURI($uri) *[line 52]*
Function Parameters:

- *string* **$uri**

Set request uri

- **Access** public

boolean function dbdRequest::__isset($key) *[line 109]*
Function Parameters:

- *string* **$key**

Check to see if a property is set

- **Access** public

Class dbdRouter

[line 17]

Request routing class

- **Package** dbdMVC
- **Usedby** dbdXajax
- **Usedby** dbdMVC
- **Usedby** dbdDispatcher
- **Uses** dbdURI
- **Uses** dbdRequest

dbdRouter::$action

string = null *[line 37]*

Action name

- **Access** private

dbdRouter::$baseUrl

string = null *[line 42]*

Base url with controller and action

- **Access** private

dbdRouter::$controller

string = null *[line 32]*

Controller name

- **Access** private

dbdRouter::$params

array = array() *[line 47]*

Key value pairs of parameters

- **Access** private

dbdRouter::$request

dbdRequest = null *[line 27]*

Instance of dbdRequest

- **Access** private

Constructor *void* function dbdRouter::__construct([$request = null]) *[line 53]*
Function Parameters:

- *dbdRequest* **$request**

Set request object, parse request uri, and build base url

- **Access** public

mixed function dbdRouter::addParam($name, $value) *[line 130]*
Function Parameters:

- *string* **$name**

- *mixed* **$value**

Add value to array parameter

- **Access** public

void function dbdRouter::buildURL() *[line 77]*

Build base url

- **Access** private

string function dbdRouter::getAction() *[line 153]*

Get action name

- **Access** public

string function dbdRouter::getController() *[line 145]*

Get controller name

- **Access** public

mixed function dbdRouter::getParam($name) *[line 162]*

Function Parameters:

- *string* **$name**

Get named parameter

- **Access** public

array function dbdRouter::getParams() *[line 173]*

Get all paramters

- **Access** public

string function dbdRouter::getURL([$get_params = false]) *[line 186]*

Function Parameters:

- *bollean* **$get_params**

Get current url.

If optional paramter is passed, all current paramters are included in the url.

- **Access** public

void function dbdRouter::parseRequest() *[line 66]*

Parse request uri

- **Access** private

string function dbdRouter::setAction($action) *[line 98]*
Function Parameters:

- *string* **$action**

Set action name

- **Access** public

string function dbdRouter::setController($controller) *[line 87]*
Function Parameters:

- *string* **$controller**

Set controller name

- **Access** public

mixed function dbdRouter::setParam($name, $value) *[line 110]*
Function Parameters:

- *string* **$name**
- *mixed* **$value**

Set named parameter

- **Access** public

void function dbdRouter::unsetParam($name) *[line 119]*
Function Parameters:

- *string* **$name**

Unset named parameter

- **Access** public

Class dbdSmarty

[line 22]

dbdMVC Smarty subclass

- **Package** dbdMVC
- **TODO** move custom plugins to seperate directory and then register them
- **Usedby** dbdController
- **Uses** Smarty
- **Uses** dbdMVC

dbdSmarty::CACHE_DIR

= ".cache" *[line 43]*

Path to directory where smarty cache will live relative to view dir.

dbdSmarty::COMPILED_TEMPLATE_DIR

= ".templates_c" *[line 35]*

Path to directory where smarty complied templates will live relative to view dir.

dbdSmarty::CONFIG_DIR

= "configs" *[line 39]*

Path to directory where smarty configs will live relative to view dir.

dbdSmarty::TEMPLATE_DIR

= "templates" *[line 31]*

Path to directory where smarty templates will live relative to view dir.

dbdSmarty::VIEW_DIR

= "views" *[line 27]*

Path to directory where smarty templates will live relative to application dir.

dbdSmarty::$app_dir

string = null *[line 73]*

Application directory

- **Access** private

dbdSmarty::$beautify_regex

array = array(
"newlines" => array("/(<[^!\/][^>]*>)(<[^!\/][^>]*>)/", "\\1\n\\2")/*,
"tabs" => array("/(\n|\t)+/", "")*/) *[line 60]*

Array of regex pairs for beautify the xhtml.

- **Access** private

dbdSmarty::$css_files

array = array() *[line 88]*

List of css files for inclusion

- **Access** private

dbdSmarty::$css_host

string = null *[line 83]*

Css host name if not DOCROOT directory

- **Access** private

dbdSmarty::$css_vars

array = array() *[line 93]*

List of css vars for inclusion

- **Access** private

dbdSmarty::$debug

boolean = false *[line 78]*

Flag to turn off minification.

- **Access** private

dbdSmarty::$flash_loader

FlashLoaderSWFO = null *[line 113]*

FlashLoader Object for including flash movies.

- **Access** private

dbdSmarty::$js_files

array = array() *[line 103]*

List of js files for inclusion

- **Access** private

dbdSmarty::$js_host

string = null *[line 98]*

JS host name if not DOCROOT directory

- **Access** private

dbdSmarty::$js_vars

array = array() *[line 108]*

List of js vars for inclusion

- **Access** private

dbdSmarty::$minify_regex

```
array = array(
      "spaces" => array("/ [ ]+/", "
"),"printable_tabs"=>array("%(?<=[^<>])\t(?=[^<>]*</(textarea|code|pre)>)%i","&#09;"),"printable_newlines"=>
array("%(?<=[^<>])\r?\n(?=[^<>]*</(textarea|code|pre)>)%i","&#10;"),"printable_whitespace"=>array("/(?<=[^<
>])(\n|\t)+(?=[^<>])/"," "),"whitespace"=>array("/(\n|\t)+/",""),"comments"=>array("/<!--.*?-->/","")) [line 48]
```

Array of regex pairs for minifying the xhtml.

- **Access** private

dbdSmarty::$qt_loader

QTLoader = null *[line 118]*

QTLoader Object for including quick time movies.

- **Access** private

dbdSmarty::$rendered

boolean = false *[line 68]*

Flag to designate if the template has been rendered.

- **Access** private

Constructor *void* function dbdSmarty::__construct($app_dir) *[line 122]*
Function Parameters:

- **$app_dir**

Smarty contructor that sets implimentation specific variables

- **Access** public

void function dbdSmarty::addCSS($files) *[line 207]*
Function Parameters:

- *mixed* **$files**

Add CSS files to the list

- **Access** public

void function dbdSmarty::addCSSVar($name, $value) *[line 218]*
Function Parameters:

- *string* **$name**
- *mixed* **$value**

Add CSS variable to the list

- **Access** public

void function dbdSmarty::addFlashMovie(0) *[line 306]*

Function Parameters:

- *mixed* **0**

Add a flash movie to FlashLoader

- **Access** public

void function dbdSmarty::addJS($files) *[line 250]*

Function Parameters:

- *mixed* **$files**

Add JS files to the list

- **Access** public

void function dbdSmarty::addJSVar($name, $value) *[line 270]*

Function Parameters:

- *string* **$name**
- *mixed* **$value**

Get a JS variable from the list

- **Access** public

void function dbdSmarty::addQTMovie(0) *[line 337]*

Function Parameters:

- *mixed* **0**

Add a flash movie to FlashLoader

- **Access** public

void function dbdSmarty::clearCSS() *[line 234]*

Clear CSS file list

- **Access** public

void function dbdSmarty::clearJS() *[line 277]*

Clear JS file list

- **Access** public

string function dbdSmarty::dbdBeautify($tpl, $smarty) *[line 424]*

Function Parameters:

- *string* **$tpl**
- *object* **$smarty**

Beautify buffer.

- **Access** public

string function dbdSmarty::dbdIncludeFiles($tpl, $smarty) *[line 357]*
Function Parameters:

- *string* **$tpl**
- *object* **$smarty**

Include CSS & JS Files Automatically

- **Access** public

string function dbdSmarty::dbdMinify($tpl, $smarty) *[line 436]*
Function Parameters:

- *string* **$tpl**
- *object* **$smarty**

Minify buffer.

- **Access** public

string function dbdSmarty::dbdTag($tpl, $smarty) *[line 448]*

Function Parameters:

- *string* **$tpl**
- *object* **$smarty**

Smarty output filter to add dbdMVC credits to source

- **Access** public

void function dbdSmarty::display($resource_name, [$cache_id = null], [$compile_id = null], [$no_includes = false]) *[line 168]*

Function Parameters:

- *string* **$resource_name**
- *string* **$cache_id**
- *string* **$compile_id**
- **$no_includes**

Executes & displays the template results.

Also sets rendered flag.

- **Access** public

void function dbdSmarty::fetch($resource_name, [$cache_id = null], [$compile_id = null], [$display = false], [$no_includes = false], [$no_minify = false]) *[line 182]*

Function Parameters:

- *string* **$resource_name**
- *string* **$cache_id**
- *string* **$compile_id**

- *boolean* **$display**
- *boolean* **$no_includes**
- *boolean* **$no_minify**

Executes & returns the template result.

- **Access** public

mixed function dbdSmarty::getCSSVar($name) *[line 227]*
Function Parameters:

- *string* **$name**

Get a CSS variable from the list

- **Access** public

mixed function dbdSmarty::getJSVar($name) *[line 261]*
Function Parameters:

- *string* **$name**

Add JS variable to the list

- **Access** public

void function dbdSmarty::initFlash(0) *[line 285]*

Function Parameters:

- *mixed* **0**

Initialize FlashLoaderSWFO

- **Access** public

void function dbdSmarty::initQuickTime(0) *[line 316]*

Function Parameters:

- *mixed* **0**

Initialize QTLoader

- **Access** public

void function dbdSmarty::setCssHost($host) *[line 199]*

Function Parameters:

- *string* **$host**

Set CSS host for inclusion

- **Access** public

void function dbdSmarty::setJSHost($host) *[line 242]*

Function Parameters:

- *string* **$host**

Set JS host for inclusion

- **Access** public

void function dbdSmarty::setOptions() *[line 134]*

Set Smarty options.

This is where all directories are set, and filters are registered.

- **Access** private

bollean function dbdSmarty::wasRendered() *[line 347]*

Has a trmplate been rendered?

- **Access** public

Class dbdURI

[line 15]

dbdMVC URI Parsing and creating class

- **Package** dbdMVC
- **Usedby** dbdRouter

dbdURI::$action

string = null *[line 35]*

Action name

- **Access** private

dbdURI::$controller

string = null *[line 30]*

Controller name

- **Access** private

dbdURI::$instance

mixed = null *[line 25]*

Active instance of dbdURI Singleton

- **Static Variable Used** object 0:
- **Static**
- **Access** private

dbdURI::$params

array = array() *[line 40]*

Key value pairs of parameters

- **Access** private

dbdURI::$position_lists

array = array() *[line 45]*

Parameter key lists, organized by controller by action

- **Access** private

string function dbdURI::create([$controller = null], [$action = null], [$args = array()]) *[line 158]*
Function Parameters:

- *string* **$controller**
- *string* **$action**
- *associative* **$args** array $args

Create a new url based on the given components

- **Static**
- **Access** public

string function dbdURI::getAction() *[line 139]*

Get action name

- **Static**
- **Access** public

string function dbdURI::getController() *[line 131]*

Get controller name

- **Static**
- **Access** public

object instance function dbdURI::getInstance() *[line 55]*

Get the active instance or create one.

- **Static**
- **Access** public

array function dbdURI::getParams() *[line 147]*

Get all parameter

- **Static**
- **Access** public

string function dbdURI::replace([$baseUri = ""], [$controller = ""], [$action = ""], [$args = array()]) *[line 207]*

Function Parameters:

- *string* **$baseUri**
- *string* **$controller**
- *string* **$action**
- *associative* **$args** array $args

Replace the given url with the new components given

- **Static**
- **Access** public

void function dbdURI::set($uri) *[line 65]*

Function Parameters:

- *string* **$uri**

Set request uri

- **Static**
- **Access** public

void function dbdURI::setParam($name, $value) *[line 91]*

Function Parameters:

- *string* **$name**
- *mixed* **$value**

Set request parameter, recursively

- **Static**

- **Access** public

void function dbdURI::setPositionList($controller, $action, [$params = array()]) *[line 120]*

Function Parameters:

- *string* **$controller**
- *string* **$action**
- *array* **$params**

Set a parameter position list for a specific controller/action.

This allows for key-less parameritized urls

- **Static**
- **Access** public

Class dbdXajax

[line 22]

dbdMVC Xajax subclass

- **Package** dbdMVC
- **Usedby** dbdController
- **Uses** xajax
- **Uses** dbdRouter
- **Uses** dbdDispatcher

dbdXajax::$router

object = null *[line 29]*

Instance of dbdRouter

- **Access** private

Constructor *void* function dbdXajax::__construct($router, [$sRequestURI = ""], [$sWrapperPrefix = dbdDispatcher::XAJAX_ACTION_PREFIX], [$sEncoding = XAJAX_DEFAULT_CHAR_ENCODING], [$bDebug = false]) *[line 38]*

Function Parameters:

- *dbdRouter* **$router**
- *string* **$sRequestURI**
- *string* **$sWrapperPrefix**
- *string* **$sEncoding**
- *boolean* **$bDebug**

Set router and construct xajax object

- **Access** public

string function dbdXajax::_xmlToArray($rootTag, $sXml) *[line 50]*

Function Parameters:

- *string* **$rootTag**
- *string* **$sXml**

Wrapper for xajax::_xmlToArray() to set paramaters with dbdRouter

- **Access** public

Class dbdXajaxJS

[line 17]

Controller class for serving xajax javascript file.

- **Package** dbdMVC
- **Uses** dbdJS
- **Uses** dbdException

void function dbdXajaxJS::doDefault() *[line 23]*

Serve xajax javascript file

- **Throws** dbdException
- **Access** public

Class FB

[line 66]

- **Package** dbdMVC

true function FB::dump($Key, $Variable) *[line 231]*

Function Parameters:

- *string* **$Key**
- *mixed* **$Variable**

Dumps key and variable to firebug server panel

- **See** FirePHP::DUMP
- **Static**
- **Throws** Exception
- **Access** public

true function FB::error($Object, [$Label = null]) *[line 218]*

Function Parameters:

- *mixes* **$Object**
- *string* **$Label**

Log object with label to firebug console

- **See** FirePHP::ERROR
- **Static**
- **Throws** Exception
- **Access** public

boolean function FB::getEnabled() *[line 86]*

Check if logging is enabled

- **See** FirePHP->getEnabled()
- **Static**
- **Access** public

array function FB::getOptions() *[line 124]*

Get options for the library

- **See** FirePHP->getOptions()
- **Static**
- **Access** public

true function FB::group($Name, [$Options = null]) *[line 155]*

Function Parameters:

- *string* **$Name**
- *array* **$Options** OPTIONAL Instructions on how to log the group

Start a group for following messages

Options: Collapsed: [true|false] Color: [#RRGGBB|ColorName]

- **Static**
- **Access** public

true function FB::groupEnd() *[line 166]*

Ends a group you have started before

- **Throws** Exception
- **Static**
- **Access** public

true function FB::info($Object, [$Label = null]) *[line 192]*

Function Parameters:

- *mixes* **$Object**
- *string* **$Label**

Log object with label to firebug console

- **See** FirePHP::INFO
- **Static**
- **Throws** Exception
- **Access** public

true function FB::log($Object, [$Label = null]) *[line 179]*

Function Parameters:

- *mixes* **$Object**
- *string* **$Label**

Log object with label to firebug console

- **See** FirePHP::LOG
- **Static**
- **Throws** Exception
- **Access** public

true function FB::send($Object) *[line 137]*

Function Parameters:

- *mixed* **$Object**

Log object to firebug

- **See** http://www.firephp.org/Wiki/Reference/Fb
- **Static**
- **Throws** Exception

- **Access** public

void function FB::setEnabled($Enabled) *[line 75]*
Function Parameters:

- *boolean* **$Enabled** TRUE to enable, FALSE to disable

Enable and disable logging to Firebug

- **See** FirePHP->setEnabled()
- **Static**
- **Access** public

void function FB::setObjectFilter($Class, $Filter) *[line 101]*
Function Parameters:

- *string* **$Class** The class name of the object
- *array* **$Filter** An array or members to exclude

Specify a filter to be used when encoding an object

Filters are used to exclude object members.

- **See** FirePHP->setObjectFilter()
- **Static**
- **Access** public

void function FB::setOptions($Options) *[line 113]*
Function Parameters:

- *array* **$Options** The options to be set

Set some options for the library

- **See** FirePHP->setOptions()
- **Static**
- **Access** public

true function FB::table($Label, $Table) *[line 256]*

Function Parameters:

- *string* **$Label**
- *string* **$Table**

Log a table in the firebug console

- **See** FirePHP::TABLE
- **Static**
- **Throws** Exception
- **Access** public

true function FB::trace($Label) *[line 243]*

Function Parameters:

- *string* **$Label**

Log a trace in the firebug console

- **See** FirePHP::TRACE
- **Static**
- **Throws** Exception
- **Access** public

true function FB::warn($Object, [$Label = null]) *[line 205]*

Function Parameters:

- *mixes* **$Object**
- *string* **$Label**

Log object with label to firebug console

- **See** FirePHP::WARN
- **Static**
- **Throws** Exception
- **Access** public

Class Index

[line 16]

dbdMVC Index Sample Controller Class

- **Package** dbdMVC
- **Uses** dbdController

void function Index::doDefault() *[line 21]*

Set a welcome message for the view

- **Access** public

Class wavEdit

[line 27]

- **Package** dbdMVC

wavEdit::$afiles

mixed = array() *[line 28]*

wavEdit::$arrData

mixed = array() *[line 29]*

wavEdit::$bdColor

mixed = *[line 38]*

wavEdit::$bgColor

mixed = *[line 36]*

wavEdit::$Db

mixed = *[line 42]*

wavEdit::$gridColor

mixed = *[line 39]*

wavEdit::$grpColor

mixed = *[line 37]*

wavEdit::$header

mixed = "" *[line 32]*

wavEdit::$imgBorder

mixed = *[line 40]*

wavEdit::$imgHeight

mixed = *[line 35]*

wavEdit::$imgWidth

mixed = *[line 34]*

wavEdit::$visual_graph_mode

mixed = *[line 41]*

wavEdit::$wavs

mixed = array() *[line 31]*

Constructor *void* function wavEdit::wavEdit() *[line 44]*
void function wavEdit::cropFiles($arr) *[line 205]*
Function Parameters:

- **$arr**

void function wavEdit::doClip($val, $min, $max) *[line 432]*
Function Parameters:

- **$val**
- **$min**
- **$max**

void function wavEdit::doCrop($pos, $time) *[line 218]*
Function Parameters:

- **$pos**
- **$time**

void function wavEdit::doDrawImg($output) *[line 498]*
Function Parameters:

- **$output**

void function wavEdit::doEcho($bytes, $echo, $inte) *[line 352]*
Function Parameters:

- **$bytes**
- **$echo**
- **$inte**

void function wavEdit::doFade($pos, $bytes) *[line 274]*
Function Parameters:

- **$pos**
- **$bytes**

void function wavEdit::doJoin() *[line 460]*
void function wavEdit::doSetVol8Bit($val) *[line 408]*
Function Parameters:

- **$val**

void function wavEdit::doSetVol16Bit($val) *[line 419]*
Function Parameters:

- **$val**

void function wavEdit::doViewHeader($head) *[line 650]*
Function Parameters:

- **$head**

void function wavEdit::drawImgWave([$val = ""]) *[line 483]*
Function Parameters:

- **$val**

void function wavEdit::getBytes($val) *[line 245]*
Function Parameters:

- **$val**

void function wavEdit::getDataArr($arr) *[line 186]*
Function Parameters:

- **$arr**

void function wavEdit::getFileName($path) *[line 85]*

Function Parameters:

- **$path**

void function wavEdit::getFiles($arr) *[line 68]*

Function Parameters:

- **$arr**

void function wavEdit::getHeader() *[line 136]*
void function wavEdit::getStructure($binstr) *[line 105]*

Function Parameters:

- **$binstr**

void function wavEdit::joinFiles([$fname = ""]) *[line 443]*

Function Parameters:

- **$fname**

void function wavEdit::setEcho($arr) *[line 329]*

Function Parameters:

- **$arr**

void function wavEdit::setFade($arr) *[line 259]*

Function Parameters:

- **$arr**

void function wavEdit::setVolume($val, [$fname = ""]) *[line 390]*

Function Parameters:

- **$val**
- **$fname**

void function wavEdit::unpackData($strbin) *[line 176]*

Function Parameters:

- **$strbin**

void function wavEdit::unpackHeader($str, [$opt = ""]) *[line 165]*

Function Parameters:

- **$str**
- **$opt**

void function wavEdit::viewHeader([$fname = ""]) *[line 632]*

Function Parameters:

- **$fname**

void function wavEdit::viewImage([$name = ""], [$opt = ""]) *[line 589]*

Function Parameters:

- **$name**
- **$opt**

void function wavEdit::writeWavFile([$val = ""]) *[line 616]*

Function Parameters:

- **$val**

Appendices

Appendix A - Class Trees

Package dbdCommon

FlashLoaderSWFO

- FlashLoaderSWFO

GDColor

- GDColor

InputCSS

- InputCSS

QTLoader

- QTLoader

wmArrays

- wmArrays

wmFileSystem

- wmFileSystem

wmSort

- wmSort

wmString

- wmString

Package dbdMVC

dbdController

- dbdController
  - dbdCSS
  - dbdEmptyController
  - dbdError
  - dbdInfo
  - dbdJS
    - dbdXajaxJS

  - Index

dbdDB

- dbdDB

dbdDispatcher

- dbdDispatcher

dbdException

- dbdException
  - dbdHoldableException

dbdLoader

- dbdLoader

dbdModel

- dbdModel

dbdMVC

- dbdMVC

dbdRequest

- dbdRequest

dbdRouter

- dbdRouter

dbdSmarty

- Smarty
  - dbdSmarty

dbdURI

- dbdURI

dbdXajax

- xajax
  - dbdXajax

FB

- FB

wavEdit

- wavEdit

Package JSMin

JSMin

- JSMin

JSMinException

- Exception
  - JSMinException

Package FirePHP

FirePHP

- FirePHP

Appendix D - Todo List

In Package dbdMVC

In **dbdController::$xajaxResponse**

- make dbdXajaxResponse object

In **dbdController**

- Use Response object for output.

In **dbdMVC.php**

- Create readme and update dbdMVC description

In **dbdSmarty**

- move custom plugins to seperate directory and then register them

In **dbdRequest::getCookie()**

- How to retrieve from nested arrays

In **dbdRequest::getPost()**

- How to retrieve from nested arrays

In **dbdRequest::getQuery()**:

- How to retrieve from nested arrays

In **dbdCSS::IMPORT_REGEX**:

- allow for url() syntax

In **dbdController::registerXajaxActions()**:

- Auto insert JS into header of view output using some filter.

Index

C

constructor dbdMVC::__construct() . . . 175

Constructer sets exception handler and the default app_name.

constructor dbdModel::__construct() . . . 164

The constructer's job is to set the table info,
get the db, and call the initializer.

constructor dbdLoader::__construct() . . . 158

Initialize the loader
<b>Note:</b> The cunstructer is private because this class can only
be instantiated by dbdLoader::getInstance() to achieve a Singleton.

constructor dbdException::__construct() . . . 147

Pass paremeters on to parent exception with prefixed message.

constructor dbdRequest::__construct() . . . 183

Prepare the request object for use by setting the request uri.

constructor dbdRouter::__construct() . . . 190

Set request object, parse request uri, and build base url

constructor wavEdit::wavEdit() . . . 221

constructor dbdXajax::__construct() . . . 212

Set router and construct xajax object

constructor dbdSmarty::__construct() . . . 199

Smarty contructor that sets implimentation specific variables

constructor dbdDispatcher::__construct() . . . 143

Set router, app_dir, and controller_dir.

constructor dbdDB::__construct() . . . 137

The constructors job is to instantiate the needed PDO object and set attributes

constructor JSMin::__construct() . . . 77

constructor FirePHP::__construct() . . . 60

The object constructor

constructor QTLoader::__construct() . . . 29

The purpose of the constructor is to initialize the main
attribute list and then fill it with the arguments (if any).

constructor GDColor::__construct() . . . 19

The constructor's job is to convert HEX color to its integer parts
and then allocate the color.

colorizeImage() . . . 106

cropImage() . . . 106

constructor dbdController::__construct() . . . 112

Prepare controller for action.

cropResizeImage() . . . 106

cropImageReal() . . . 106

constructor FlashLoaderSWFO::__construct() . . . 13

The purpose of the constructor is to initialize the main
attribute list and then fill it with the arguments (if any).

D

dbdModel::__set() . . . 170
Magic function for setting field values

dbdModel::__unset() . . . 170
Magic function to unset a field

dbdModel::__isset() . . . 170
Magic function to check if a field is set

dbdModel::__get() . . . 169
Magic function for getting field values

dbdModel::__call() . . . 169
Magic function allows calling of extra magic functions for testing field values.

dbdMVC . . . 171
dbdMVC Front Controller Class

dbdMVC::VERSION . . . 171
Current version number

dbdMVC::$dispatcher . . . 172
Instance of dbdDispatcher

dbdMVC::$debug_mode . . . 172
Current debug mode.

dbdMVC::$app_name . . . 172
Name of dbdApp

dbdMVC::$app_dir . . . 171
Path to dbdApp files (models, views, and controllers dirs)

dbdModel::setDB() . . . 169
Set the database object

dbdModel::save() . . . 168
Save all the fields for this row

dbdModel::getConstant() . . . 166
Return the value of a class constant using Reflection

dbdModel::getCount() . . . 166
Get a count of all rows from this table

dbdModel::getAll() . . . 166
Get all rows from this table

dbdModel::ensureDB() . . . 165
Make sure we have an instance of the db

dbdModel::delete() . . . 165
Delete this row

dbdModel::getData() . . . 167
Get the fields for this row

dbdModel::getDB() . . . 167
Get the database object

dbdModel::initFields() . . . 168
Select all the fields names for this table

dbdModel::init() . . . 168
Select all the fields for this row

dbdModel::getReflection() . . . 168
Return the Reflection of a class

dbdModel::getID() . . . 167
Get the row id

dbdMVC::$error_controller . . . 173
Error/Default controller (if left null, the dbdDispatcher::ERROR_CONTROLLER will be used)

dbdMVC::$error_log . . . 173

Error log file (if left null, system default will be used)
dbdMVC::getExecutionTime() 178
Calculate execution time of the application and return.
dbdMVC::getFallbackController() 178
Get fallback controller name.
dbdMVC::getErrorLog() 177
Return the error_log file name.
dbdMVC::getErrorController() 177
Get error controller name.
dbdMVC::getAppName() 177
Get application name.
dbdMVC::getInstance() 178
Get the active instance of dbdMVC or create one.
dbdMVC::getRequest() 178
Get the instance of dbdRequest.
dbdMVC::PHPisExposed() 179
Check if phpinfo display is allowed.
dbdMVC::logExecutionTime() 179
Log execution time of the application to the error_log.
dbdMVC::init() 179
Prepare for dispatch by setting up the dbdDispatcher
and by adding the application specific paths to dbdLoader.
dbdMVC::getRouter() 178
Get the instance of dbdRouter.
dbdMVC::getAppDir() 177
Get application directory.
dbdMVC::genCLIURI() 176
Generate a URI from the Command Line Arguments.
dbdMVC::$request 174
Instance of dbdRequest
dbdMVC::$instance 174
Active instance of dbdMVC Singleton
dbdMVC::$fallback_controller 173
Fallback/Default controller (if left null, the dbdDispatcher::FALLBACK_CONTROLLER will be used)
dbdMVC::$expose_php_info 173
Allow phpinfo() via dbdInfo controller
dbdMVC::$router 174
Instance of dbdRouter
dbdMVC::$start 175
Start time of dbdApp, from microtime()
dbdMVC::exposePHPInfo() 176
Allow phpinfo() display via dbdInfo.
dbdMVC::e() 176
Default ecpetion handler
dbdMVC::dispatch() 175
Dispatch dbdDispatcher.
dbdMVC::debugMode() 175
Check if debug mode is set.
dbdModel::buildWhereClause() 165
Build where clause for count and getAll methods
dbdModel::$table_name 164
The name of the table

dbdJS::init() 155
Set no render, cache_dir, debug, and load JSMin
dbdJS::minify() 155
Minify buffer.
dbdJS::genURL() 154
Generate a combine url from an array of files
dbdJS::genCacheName() 154
Generate a cache file name based on the list of proccess files.
dbdJS::ensureResource() 154
Ensure a resource is open and valid.
dbdJS::output() 155
Set headers, minify, and echo buffer.
dbdJS::readFiles() 155
Read and close a file.
dbdLoader::DEFAULT_MODEL_DIR 156
Model directory
dbdLoader::DEFAULT_CONTROLLER_DIR 156
Controller directory
dbdLoader 156
dbdMVC File and Class Loader
dbdJS::setHeaders() 156
Set js headers
dbdJS::doGet() 153
Serve js files...
dbdJS::doDefault() 153
Alias of doGet()
dbdJS::$debug 151
Flag to turn off minification.
dbdJS::$files 152
List of files for proccessing
dbdJS::$cache_file 151
Cache file name
dbdJS::$cache_dir 151
Cache directory
dbdJS::$buffer 150
Output buffer that will be minified if debug is off.
dbdJS::$vars 152
List of external varables to be added as
properties of the dbdJS javascript object.
dbdJS::addFile() 152
Check a file for existence and then
add it to the array for later proccessing.
dbdJS::doCombine() 153
Serve multiple js files as one
dbdJS::createCache() 153
Create cache file and dump buffer.
dbdJS::checkCache() 153
Check for a cache file and make sure its not outdated.
dbdJS::addVars() 152
Add external variables to the buffer.
dbdLoader::$controller_dirs 157
Controller directory
dbdLoader::$include_paths 157

Array of directories to find classes and other files

dbdModel 162

dbdMVC Database Table Model

dbdModel::CONST_TABLE_KEY 163

dbdLoader::setModelDir() 162

Set model directory

dbdLoader::setControllerDir() 161

Set controller directory

dbdLoader::search() 161

Search for a file in the list of path(s).

dbdModel::CONST_TABLE_NAME 163

dbdModel::$data 163

Array of fields from row

dbdModel::$table_key 164

The key for the table

dbdModel::$reflections 164

A list of class reflections to limit overhead

dbdModel::$id 163

Row id

dbdModel::$db 163

Static instance of dbdDB

dbdLoader::register() 161

Add path to array and to php.ini

dbdLoader::loadClass() 160

Load a class file. Calls dbdLoader::load().

dbdLoader::addPath() 158

Add path to dbdLoader and php.ini

dbdLoader::addControllerDir() 158

Add a controller directory

dbdLoader::$model_dir 157

Model directory

dbdLoader::$instance 157

Active instance of dbdLoader Singleton

dbdLoader::getControllerDirs() 159

Get controller directory

dbdLoader::getInstance() 159

Get the active instance or create one.

dbdLoader::load() 160

Load a php file using require_once if not already loaded.

dbdLoader::init() 160

Register php.ini include_path(s), controller dir, and model dir

dbdLoader::getPaths() 159

Get all paths registered with dbdLoader

dbdLoader::getModelDir() 159

Get model directory

dbdMVC::run() 179

Run dbdMVC on the given dbdApp dir.

dbdMVC::setAppDir() 180

Set application directory property.

dbdSmarty::dbdIncludeFiles() 202

Include CSS & JS Files Automatically

dbdSmarty::dbdMinify() 202

Minify buffer.

dbdSmarty::dbdBeautify() 201
Beautify buffer.
dbdSmarty::clearJS() 201
Clear JS file list
dbdSmarty::clearCSS() 201
Clear CSS file list
dbdSmarty::dbdTag() 203
Smarty output filter to add dbdMVC credits to source
dbdSmarty::display() 203
Executes & displays the template results.
dbdSmarty::initFlash() 205
Initialize FlashLoaderSWFO
dbdSmarty::getJSVar() 204
Add JS variable to the list
dbdSmarty::getCSSVar() 204
Get a CSS variable from the list
dbdSmarty::fetch() 203
Executes & returns the template result.
dbdSmarty::addQTMovie() 201
Add a flash movie to FlashLoader
dbdSmarty::addJSVar() 200
Get a JS variable from the list
dbdSmarty::$js_vars 197
List of js vars for inclusion
dbdSmarty::$minify_regex 198
Array of regex pairs for minifying the xhtml.
dbdSmarty::$js_host 197
JS host name if not DOCROOT directory
dbdSmarty::$js_files 197
List of js files for inclusion
dbdSmarty::$flash_loader 197
FlashLoader Object for including flash movies.
dbdSmarty::$qt_loader 198
QTLoader Object for including quick time movies.
dbdSmarty::$rendered 198
Flag to designate if the template has been rendered.
dbdSmarty::addJS() 200
Add JS files to the list
dbdSmarty::addFlashMovie() 200
Add a flash movie to FlashLoader
dbdSmarty::addCSSVar() 199
Add CSS variable to the list
dbdSmarty::addCSS() 199
Add CSS files to the list
dbdSmarty::initQuickTime() 205
Initialize QTLoader
dbdSmarty::setCssHost() 205
Set CSS host for inclusion
dbdURI::set() 210
Set request uri
dbdURI::setParam() 210
Set request parameter, recursively
dbdURI::replace() 209

Replace the given url with the new components given

dbdURI::getParams() . . . 209

Get all parameter

dbdURI::getInstance() . . . 209

Get the active instance or create one.

dbdURI::setPositionList() . . . 211

Set a parameter position list for a specific controller/action.

dbdXajax . . . 211

dbdMVC Xajax subclass

dbdXajaxJS::doDefault() . . . 213

Serve xajax javascript file

dbdXajaxJS . . . 213

Controller class for serving xajax javascript file.

dbdXajax::_xmlToArray() . . . 212

Wrapper for xajax::_xmlToArray() to set paramaters with dbdRouter

dbdXajax::$router . . . 211

Instance of dbdRouter

dbdURI::getController() . . . 209

Get controller name

dbdURI::getAction() . . . 208

Get action name

dbdURI . . . 206

dbdMVC URI Parsing and creating class

dbdSmarty::wasRendered() . . . 206

Has a trmplate been rendered?

dbdSmarty::setOptions() . . . 206

Set Smarty options.

dbdSmarty::setJSHost() . . . 206

Set JS host for inclusion

dbdURI::$action . . . 207

Action name

dbdURI::$controller . . . 207

Controller name

dbdURI::create() . . . 208

Create a new url based on the given components

dbdURI::$position_lists . . . 208

Parameter key lists, organized by controller by action

dbdURI::$params . . . 208

Key value pairs of parameters

dbdURI::$instance . . . 207

Active instance of dbdURI Singleton

dbdSmarty::$debug . . . 196

Flag to turn off minification.

dbdSmarty::$css_vars . . . 196

List of css vars for inclusion

dbdRequest::has() . . . 187

Alias to __isset()

dbdRequest::isXmlHttpRequest() . . . 187

Is the request a Javascript XMLHttpRequest?
Should work with Prototype/Script.aculo.us, possibly others.

dbdRequest::getServer() . . . 186

Retrieve a member of the $_SERVER superglobal.

dbdRequest::getRequestURI() . . . 186

Get request uri

dbdRequest::getQuery() 186

Retrieve a member of the $_GET superglobal.

dbdRequest::set() 187

Set parameters at runtime

dbdRequest::setRequestURI() 188

Set request uri

dbdRouter::$baseUrl 189

Base url with controller and action

dbdRouter::$action 189

Action name

dbdRouter 188

Request routing class

dbdRequest::__isset() 188

Check to see if a property is set

dbdRequest::getPost() 185

Retrieve a member of the $_POST superglobal.

dbdRequest::getHeader() 185

Return the value of the given HTTP header. Pass the header name as the plain, HTTP-specified header name. Ex.: Ask for 'Accept' to get the Accept header, 'Accept-Encoding' to get the Accept-Encoding header.

dbdMVC::setErrorLog() 181

Set error_log property.

dbdMVC::setFallbackController() 182

Set fallback_controller property.

dbdMVC::setErrorController() 181

Set error_controller property.

dbdMVC::setDebugMode() 181

Set debug mode.

dbdMVC::setAppName() 180

Set application name property.

dbdRequest 182

HTTP Request Class.

dbdRequest::$params 183

Additional parameters set at runtime since the $_SUPER_GLOBAL's seem to be read only.

dbdRequest::getEnv() 184

Retrieve a member of the $_ENV superglobal.

dbdRequest::getCookie() 184

Retrieve a member of the $_COOKIE superglobal.

dbdRequest::get() 183

Access values contained in the superglobals as public members

dbdRequest::$request_uri 183

HTTP request URI

dbdRouter::$controller 189

Controller name

dbdRouter::$params 190

Key value pairs of parameters

dbdSmarty::COMPILED_TEMPLATE_DIR 195

Path to directory where smarty complied templates will live relative to view dir.

dbdSmarty::CONFIG_DIR 195

Path to directory where smarty configs will live relative to view dir.

dbdSmarty::CACHE_DIR 194

Path to directory where smarty cache will live relative to view dir.

dbdSmarty . . . 194
dbdMVC Smarty subclass
dbdRouter::unsetParam() . . . 194
Unset named parameter
dbdSmarty::TEMPLATE_DIR . . . 195
Path to directory where smarty templates will live relative to view dir.
dbdSmarty::VIEW_DIR . . . 195
Path to directory where smarty templates will live relative to application dir.
dbdSmarty::$css_host . . . 196
Css host name if not DOCROOT directory
dbdSmarty::$css_files . . . 196
List of css files for inclusion
dbdSmarty::$beautify_regex . . . 195
Array of regex pairs for beautify the xhtml.
dbdSmarty::$app_dir . . . 195
Application directory
dbdRouter::setParam() . . . 193
Set named parameter
dbdRouter::setController() . . . 193
Set controller name
dbdRouter::getAction() . . . 191
Get action name
dbdRouter::buildURL() . . . 191
Build base url
dbdRouter::addParam() . . . 190
Add value to array parameter
dbdRouter::$request . . . 190
Instance of dbdRequest
dbdRouter::getController() . . . 191
Get controller name
dbdRouter::getParam() . . . 191
Get named parameter
dbdRouter::setAction() . . . 193
Set action name
dbdRouter::parseRequest() . . . 192
Parse request uri
dbdRouter::getURL() . . . 192
Get current url.
dbdRouter::getParams() . . . 192
Get all paramters
dbdJS::DIR_DELIM_REGEX . . . 150
Directory delimiter for passing a string of files
dbdJS . . . 150
Controller class for compressing and serving js files.
dbdController::init() . . . 116
Default init method.
dbdController::noRender() . . . 117
Disable auto render on destruction.
dbdController::getURL() . . . 116
Get current request url.
dbdController::getUploadInfo() . . . 115
Get the progress information on a current upload via APC.
dbdController::getTemplate() . . . 115

Get template filename.

dbdController::registerXajaxActions() 117

Register controller actions with dbdXajax.

dbdController::setParam() 117

Set request parameter.

dbdCSS 119

Controller class for compressing and serving css files.

dbdController::__destruct() 118

Conclude the controller action.

dbdController::unsetParam() 118

Unset request parameter.

dbdController::setTemplate() 118

Set template filename.

dbdController::getParams() 115

Get all parameters.

dbdController::getParam() 115

Get request parameter.

dbdController::autoRender() 113

Auto render the view.

dbdController::dnit() 113

Default dnit method.

dbdController::autoExec() 112

Auto execute method to be executed after the action.

dbdController::assignAllParams() 112

Assign all request parameters to the view as an associative array.

dbdController::$xajaxResponse 111

Instance of xajaxResponse object

dbdController::doDefault() 113

Default action method.

dbdController::doXajax() 113

Prepare controller for xajax request.

dbdController::getController() 114

Get controller name.

dbdController::getAppDir() 114

Get application directory.

dbdController::getAction() 114

Get action name.

dbdController::forward() 114

Preform an HTTP Location change.

dbdCSS::BLOCK_CLOSE_REGEX 119

Css block close pattern.

dbdCSS::BLOCK_OPEN_REGEX 119

Css block open pattern.

dbdCSS::VAR_CALL_REGEX 122

Css variable call pattern. (for dereferencing)

dbdCSS::VAR_DEF_REGEX 122

Css variable definition pattern.

dbdCSS::MATH_SAFE_REGEX 121

Css math safe characters pattern. (for simple equations)

dbdCSS::MATH_CALL_REGEX 121

Css math call pattern. (for simple equations)

dbdCSS::IMPORT_REGEX 121

Css import regular expression.

dbdCSS::VAR_TAG_REGEX . . . 122
Css variables at-tag pattern.
dbdCSS::$buffer . . . 122
Output buffer that will be minified if debug is off.
dbdCSS::$button_props_generated_global . . . 123
List of css generated button global properties.
dbdCSS::$button_props . . . 123
List of available css button properties.
dbdCSS::$buttons_generated . . . 123
List of css buttons that have been generated.
dbdCSS::$buttons . . . 122
List of css buttons that have been defined.
dbdCSS::DIR_DELIM_REGEX . . . 121
Directory delimiter for passing a string of files
dbdCSS::DEBUG_REGEX . . . 121
Debug comment to turn off minification.
dbdCSS::BUTT_PROP_DEF_DISABLED_REGEX . . . 120
Css button property disabled pattern.
dbdCSS::BUTT_GEN_TPL_TAG_REGEX . . . 120
Css generated buttons template tag pattern.
dbdCSS::BUTT_GEN_TAG_REGEX . . . 120
Css generated buttons at-tag pattern.
dbdCSS::BUTT_DEF_TAG_REGEX . . . 119
Css button definition pattern.
dbdCSS::BUTT_PROP_DEF_HOVER_REGEX . . . 120
Css button property hover pattern.
dbdCSS::BUTT_PROP_DEF_REGEX . . . 120
Css button property definition pattern.
dbdCSS::CSS_EXT_REGEX . . . 121
Css file extension pattern.
dbdCSS::CACHE_INFO_SPRITE . . . 121
Css caching info sprite pattern.
dbdCSS::CACHE_INFO_FILES . . . 120
Css caching info file list pattern.
dbdCSS::BUTT_TAG_REGEX . . . 120
Css buttons at-tag pattern.
dbdController::$xajax . . . 111
Instance of dbdXajax
dbdController::$view . . . 111
Instance of dbdSmarty
DBCONN_INCLUDED . . . 85
Constant boolean to flag file inclusion.
DB_DRIVER . . . 85
Database connection strings
DBCONN_DEFAULT_DB . . . 85
Constant default database name if none is specified at connection time
dbconn.inc . . . 85
dbconn.inc :: The Database string constants
DBD_XAJAX_DIR . . . 82
xAjax package directory
DB_HOST . . . 85
DB_PASS . . . 85
DBD_DOC_ROOT . . . 87

DBD_APP_DIR 87
DB_USER 85
DB_PORT 85
DBD_SMARTY_PLUG_DIR 82
Smarty add0n plug-in directory directory (used to avoid disrupting standard plug-in dir)
DBD_SMARTY_DIR 82
Smarty package directory
DBD_DEBUG_CSS 82
DBD_DEBUG_DB 82
Turn on all debugging
DBD_DEBUG_ALL 82
Turn on all debugging
DBD_CACHE_DIR 82
dbdMVC cache directory
dbdLog() 81
DBD_DEBUG_HTML 82
DBD_DEBUG_JS 82
DBD_PLUG_DIR 82
dbdMVC plug-in directory
DBD_MVC_CLI 82
Are we running in CLI mode?
DBD_FIREPHP_DIR 82
FirePHP core directory
DBD_DS 82
Shorthand for DIRECTORY_SEPARATOR
DBD_MVC_DIR 87
dbdController.php 88
dbdController.php :: dbdController Class File
dbdXajax.php 104
dbdXajax.php :: dbdXajax Class File
dbdXajaxJS.php 105
dbdXajaxJS.php :: dbdXajaxJS Class File
dbdURI.php 103
dbdURI.php :: dbdURI Class File
dbdSmarty.php 102
dbdSmarty.php :: dbdSmarty Class File
dbdRouter.php 101
dbdRouter.php :: dbdRouter Class File
dbdController 109
Abstract Parent Controller Class
All controller classes must extend this class or a child of this class.
dbdController::$app_dir 109
Application directory
dbdController::$template 110
Template filename
dbdController::$router 110
Instance of dbdRouter
dbdController::$model 110
Instance of some model.
dbdController::$default_render 110
Should dbdController attempt to render
dbdRequest.php 100
dbdRequest.php :: dbdRequest Class File

dbdModel.php 99

dbdModel.php :: dbdModel Class File

dbdEmptyController.php 92

dbdEmptyController.php :: dbdEmptyController Class File

dbdDispatcher.php 91

dbdDispatcher.php :: dbdDispatcher Class File

dbdDB.php 90

dbdDB.php :: dbdDB Class File

dbdCSS.php 89

dbdCSS.php :: dbdCSS Class File

dbdError.php 93

dbdError.php :: dbdError Class File

dbdException.php 94

dbdException.php :: dbdException Class File

dbdLoader.php 98

dbdLoader.php :: dbdLoader Class File & __autoload() Function

dbdJS.php 97

dbdJS.php :: dbdJS Class File

dbdInfo.php 96

dbdInfo.php :: dbdInfo Class File

dbdHoldableException.php 95

dbdHoldableException.php :: dbdHoldableException Class File

dbdCSS::$button_props_generated_tpl 124

List of css generated button template properties.

dbdCSS::$button_props_global 124

List of css button global properties.

dbdDispatcher::DEFAULT_ACTION 141

Default action name

dbdDispatcher::DEFAULT_CONTROLLER 141

Default controller name

dbdDispatcher::BUILTIN_PREFIX 141

Builtin controller class prefix

dbdDispatcher::ACTION_PREFIX 141

Action method name prefix

dbdDispatcher 141

Dispatch controller actions.

dbdDispatcher::ERROR_CONTROLLER 142

Error controller class if is set

dbdDispatcher::FALLBACK_CONTROLLER 142

Fallback controller class if none exists

dbdDispatcher::$controller_dirs 142

Controller directory

dbdDispatcher::$app_dir 142

Application directory

dbdDispatcher::XAJAX_ACTION_PREFIX 142

Xajax action method name prefix

dbdDispatcher::MAGIC_ACTION 142

Magic action name

dbdDB::__call() 140

Magic function to call PDO functions and rethrow any

dbdDB::time() 140

Format time for SQL insert

dbdDB::$pdo 137

PDO instance variable

dbdDB::assembleDSN() . . . 137
Assemble driver specific DSN's

dbdDB::$options . . . 136
Array of PDO options to set after construction

dbdDB::$instances . . . 136
array of dbdDB objects

dbdDB::TIME_FORMAT . . . 136
SQL compliant time formating string

dbdDB::date() . . . 138
Format date for SQL insert

dbdDB::getInstance() . . . 138
Include db_conn string file.

dbdDB::setAttributes() . . . 140
Set default PDO attributes

dbdDB::prepExec() . . . 139
Prepare & excute a query statement

dbdDB::nextAutoID() . . . 139
Selects the next auto_increment id

dbdDB::includeDBConn() . . . 139
Include dbconn string file.

dbdDispatcher::$fallback_controller . . . 143
User defined fallback controller

dbdDispatcher::$router . . . 143
Instance of dbdRouter

dbdHoldableException::$held . . . 148
Array of held exceptions

dbdHoldableException::$hold . . . 148
Flag holding status

dbdHoldableException . . . 147
dbdMVC Exception class.

dbdException . . . 147
dbdMVC Exception class.

dbdError::doError() . . . 146
Crazy static action method expects intstance of dbdController for dbdError extension.

dbdHoldableException::getHeld() . . . 148
Get array of held exceptions

dbdHoldableException::hold() . . . 148
Enable holding

dbdInfo::doDefault() . . . 150
Render phpinfo() if PHPisExposed or forward home.

dbdInfo . . . 149
Controller for displaying phpInfo.

dbdHoldableException::release() . . . 149
Disable holding and throw the first held excetion, if any

dbdHoldableException::intercept() . . . 149
Exception handler and method to call when ready to throw

dbdError::doDefault() . . . 146
Attept to render error.tpl or just echo error.

dbdError::ERR_500 . . . 146
HTTP Status 500 Description

dbdEmptyController . . . 144
Empty controller class for rendering templates without a controller!

dbdDispatcher::getController() 144
Get controller object.
dbdDispatcher::getAction() 144
Get action method name.
dbdDispatcher::dispatch() 143
Dispatch actions
dbdError 145
Error handling controller class.
dbdError::ERR_400 145
HTTP Status 400 Description
dbdError::ERR_408 146
HTTP Status 408 Description
dbdError::ERR_404 145
HTTP Status 404 Description
dbdError::ERR_403 145
HTTP Status 403 Description
dbdError::ERR_401 145
HTTP Status 401 Description
dbdDB::DBCONN_PATH 135
path to database connection string file
dbdDB::DATE_FORMAT 135
SQL compliant date formating string
dbdCSS::doCombine() 128
Serve multiple css files as one
dbdCSS::dumpFile() 129
Dump and close a file.
dbdCSS::createCache() 128
Create cache file and dump buffer.
dbdCSS::checkCache() 128
Check for a cache file and make sure its not outdated.
dbdCSS::checkButtonPropsRequired() 128
Check that the required property exists.
dbdCSS::ensureResource() 129
Ensure a resource is open and valid.
dbdCSS::genButton() 129
Generate a layered button from source images.
dbdCSS::genButtonSprite() 131
Generate the sprite with all of the buttons
and register their offsets in the properties array.
dbdCSS::genButtons() 130
Genrate layered buttons from source images.
dbdCSS::genButtonCSS() 130
Generate css for the defined buttons.
dbdCSS::genButtonCacheFile() 130
Generate the cached sprite file name.
dbdCSS::checkButtonCache() 127
Check for a cached button.
dbdCSS::$vars 127
List of css varables that have been defined.
dbdCSS::$debug 125
Flag to turn off minification.
dbdCSS::$cache_file 125
Cache file name

dbdCSS::$cache_dir 124
Cache directory
dbdCSS::$button_props_required 124
List of required css button properties.
dbdCSS::$file 125
Current parser file
dbdCSS::$files 126
List of files for proccessing
dbdCSS::$sprite 127
Sprite file name
dbdCSS::$minify_regex 126
Array of regex pairs for minifying the css.
dbdCSS::$line_num 126
Current parser line number
dbdCSS::$line 126
Current parser line
dbdCSS::genCacheName() 131
Generate a cache file name based on the list of proccess files.
dbdCSS::genURL() 131
Generate a combine url from an array of files
dbdDB::COMP_LT 135
dbdDB::COMP_LTEQ 135
dbdDB::COMP_LIKE 135
dbdDB::COMP_IN 135
dbdDB::COMP_GTEQ 134
dbdDB::COMP_NBETWEEN 135
dbdDB::COMP_NEQ 135
dbdDB::COMP_TYPE 135
dbdDB::COMP_NULLEQ 135
dbdDB::COMP_NLIKE 135
dbdDB::COMP_NIN 135
dbdDB::COMP_GT 134
dbdDB::COMP_EQ 134
dbdCSS::math() 132
dbdCSS::init() 132
Set debug
dbdCSS::getVar() 132
dbdCSS::getButtonProps() 132
Get properties for a button
dbdCSS::minify() 133
Minify buffer.
dbdCSS::output() 133
Set headers, minify, and echo buffer.
dbdDB::COMP_BETWEEN 134
dbdDB 134
dbdMVC Database Abstraction Layer
dbdCSS::setHeaders() 134
Set css headers
dbdCSS::parseImports() 133
Parse css files for import statements and recurse when found.
dbdMVC.php 81
dbdMVC.php :: dbdMVC Include File & Front Controller Class

F

FirePHP::log() . . . 68

Log object with label to firebug console

FirePHP::newException() . . . 68

Returns a new exception

FirePHP::json_utf82utf16() . . . 67

convert a string from one UTF-8 char to one UTF-16 char

FirePHP::json_name_value() . . . 67

array-walking function for use in generating JSON-formatted name-value pairs

FirePHP::json_encode() . . . 66

encodes an arbitrary variable into JSON format

FirePHP::registerAssertionHandler() . . . 68

Register FirePHP driver as your assert callback

FirePHP::registerErrorHandler() . . . 69

Register FirePHP as your error handler

FirePHP::setObjectFilter() . . . 70

Specify a filter to be used when encoding an object

FirePHP::setOptions() . . . 70

Set some options for the library

FirePHP::setHeader() . . . 70

Send header

FirePHP::setEnabled() . . . 69

Enable and disable logging to Firebug

FirePHP::registerExceptionHandler() . . . 69

Register FirePHP as your exception handler

FirePHP::jsonEncode() . . . 66

Encode an object into a JSON string

FirePHP::is_utf8() . . . 66

Returns true if $string is valid UTF-8 and false otherwise.

FirePHP::getEnabled() . . . 63

Check if logging is enabled

FirePHP::getInstance() . . . 64

Gets singleton instance of FirePHP

FirePHP::fb() . . . 63

Log varible to Firebug

FirePHP::exceptionHandler() . . . 63

FirePHP's exception handler

FirePHP::errorHandler() . . . 62

FirePHP's error handler

FirePHP::getOptions() . . . 64

Get options from the library

FirePHP::getUserAgent() . . . 64

Get user agent

FirePHP::init() . . . 65

Creates FirePHP object and stores it for singleton access

FirePHP::info() . . . 65

Log object with label to firebug console

FirePHP::groupEnd() . . . 65

Ends a group you have started before

FirePHP::group() . . . 64

Start a group for following messages.

FirePHP::setProcessorUrl() . . . 71

Set custom processor url for FirePHP

FirePHP::setRendererUrl() 71

Set custom renderer url for FirePHP

FB::info() 215

Log object with label to firebug console

FB::log() 216

Log object with label to firebug console

FB::groupEnd() 215

Ends a group you have started before

FB::group() 215

Start a group for following messages

FB::getOptions() 214

Get options for the library

FB::send() 216

Log object to firebug

FB::setEnabled() 217

Enable and disable logging to Firebug

FB::trace() 218

Log a trace in the firebug console

FB::warn() 219

Log object with label to firebug console

FB::table() 218

Log a table in the firebug console

FB::setOptions() 217

Set some options for the library

FB::setObjectFilter() 217

Specify a filter to be used when encoding an object

FB::getEnabled() 214

Check if logging is enabled

FB::error() 214

Log object with label to firebug console

FirePHP::_escapeTrace() 73

Escape trace path for windows systems

FirePHP::_escapeTraceFile() 73

Escape file information of trace for windows systems

FirePHP::warn() 72

Log object with label to firebug console

FirePHP::trace() 72

Log a trace in the firebug console

FirePHP::table() 72

Log a table in the firebug console

FirePHP::_standardizePath() 73

Standardizes path for windows systems.

FirePHP::__sleep() 74

When the object gets serialized only include specific object members.

FB::dump() 213

Dumps key and variable to firebug server panel

FB 213

fb() 84

Sends the given data to the FirePHP Firefox Extension.

fb.php 84

FirePHP::error() 62

Log object with label to firebug console

FirePHP::encodeTable() . . . 61

Encodes a table by encoding each row and column with encodeObject()

FlashLoaderSWFO::loadMovie() . . . 15

Loads (embbeds) flash movie in (X)HTML document.

FlashLoaderSWFO::loadMovies() . . . 16

Loads movies in list created by addMovie()

FlashLoaderSWFO::hasMovies() . . . 15

Check to see if there are any movies to load.

FlashLoaderSWFO::getMovies() . . . 15

Gets movies in list created by addMovie()

FlashLoaderSWFO::getMovie() . . . 14

Gets (embbeds) flash movie in (X)HTML document.

FlashLoaderSWFO::setAtts() . . . 16

Fill the attribute list with the arguments (if any).

FlashLoaderSWFO::_buildSWFObject() . . . 16

Builds the SWFObject JavaScript calls.

FirePHP.class.php . . . 53

**** BEGIN LICENSE BLOCK ******

FirePHP . . . 55

Sends the given data to the FirePHP Firefox Extension.

FlashLoaderSWFO::_optionalAtt() . . . 17

Checks if attribute is optional.

FlashLoaderSWFO::_missingReqAtt() . . . 17

Checks attribute list against required attribute list.

FlashLoaderSWFO::_getAtts() . . . 16

Parses attribute/value string pairs into an array and returns them combined with current attribute list.

FlashLoaderSWFO::getCommonJSPath() . . . 14

Prints include to JavaScript library.

FlashLoaderSWFO::getCommonJS() . . . 14

Prints include to JavaScript library.

FlashLoaderSWFO::$_codebase . . . 11

codebase without version number

FlashLoaderSWFO::$_common_js_shown . . . 11

flag to be set so we write JavaScript only once

FlashLoaderSWFO::$_class_id . . . 11

the flash plugin class id

FlashLoaderSWFO::$_atts . . . 11

list of main attribute defaults

FlashLoaderSWFO . . . 10

The FlashLoaderSWFO class is used to effortlessly embedd Flash movies in a (X)HTML Document.

FlashLoaderSWFO::$_js_path . . . 12

path to SWFObject external JavaScripts library

FlashLoaderSWFO::$_movies . . . 12

list of movies added my addMovie()

FlashLoaderSWFO::addMovie() . . . 14

Adds movie arguments to list of movies to be embedded later by loadMovies().

FlashLoaderSWFO::$_swfo_atts . . . 13

list of SWFObject attribute defaults

FlashLoaderSWFO::$_req_atts . . . 13

list of required attribute defaults

FlashLoaderSWFO::$_pluginspage . . . 12

plugins page url

FirePHP::DUMP 55
Dumps a variable to firebug's server panel

FirePHP::ERROR 55
Firebug ERROR level

FirePHP::$objectStack 59
A stack of objects used to detect recursion during object encoding

FirePHP::$options 59
Options for the library

FirePHP::$objectFilters 59
Filters used to exclude object members when encoding

FirePHP::$messageIndex 58
Wildfire protocol message index

FirePHP::$json_objectStack 58
Keep a list of objects as we descend into the array so we can detect recursion.

FirePHP::$throwAssertionExceptions 59
Flag whether to throw PHP assertion errors that have been converted to Exceptions

FirePHP::$throwErrorExceptions 60
Flag whether to throw PHP errors that have been converted to ErrorExceptions

FirePHP::encodeObject() 61
Encodes an object including members with
protected and private visibility

FirePHP::dump() 61
Dumps key and variable to firebug server panel

FirePHP::detectClientExtension() 60
Check if FirePHP is installed on client

FirePHP::assertionHandler() 60
FirePHP's assertion handler

FirePHP::$instance 58
Singleton instance of FirePHP

FirePHP::$inExceptionHandler 57
Flag whether we are logging from within the exception handler

FirePHP::INFO 56
Firebug INFO level

FirePHP::LOG 56
Firebug LOG level

FirePHP::GROUP_START 56
Starts a group in firebug console

FirePHP::GROUP_END 56
Ends a group in firebug console

FirePHP::EXCEPTION 56
Displays an exception in firebug console

FirePHP::TABLE 56
Displays an table in firebug console

FirePHP::TRACE 56
Displays a stack trace in firebug console

FirePHP::$enabled 57
Flag to enable/disable logging

FirePHP::$convertAssertionErrorsToExceptions 57
Flag whether to convert PHP assertion errors to Exceptions

FirePHP::WARN 57
Firebug WARN level

FirePHP::VERSION 57

FirePHP version

FlashLoaderSWFO.php 2

FlashLoaderSWFO.php :: Flash movie loading/embedding class

G

GDColor::getHex() 20

Get the HEX color value of the allocated color.

GDColor::getGreen() 20

Get the integer color value for the Green channel.

GDColor::getRed() 20

Get the integer color value for the Red channel.

GDColor::hex2arr() 20

*Static function for breaking a HEX
color into its various channels.*

GDColor::__destruct() 21

Clean up the allocated colors.

GDColor::int2arr() 21

*Static function for breaking an integer
color into its various channels.*

GDColor::getColor() 19

Get the integer color value for use in GD functions.

GDColor::getBlue() 19

Get the integer color value for the Blue channel.

GDColor::$color 18

Integer color value to be used by GD functions.

GDColor 18

*The GDColor Class is used to easily
handle colors while using the GD Library.*

GDColor::$im 18

GD Image resource to allocate colors for.

GDColor::$rgba 18

*Red, Green, Blue, and Alpha channel values
stored in an array respectively.*

GDColor::getAlpha() 19

Get the integer color value for the Alpha channel.

GDColor.php 3

GDColor.php :: The GD Color Class

I

imageCreateFrom() 106

index.php 87

index.php :: dbdMVC Bootstrap Sample File

Index.php 86

Index.php :: Index Controller Class File

InputCSS::reset() 25

Generate a reset button.

imagefilledellipsealpha() 107

imageLineThick() 107

Index::doDefault() 219

Set a welcome message for the view

Index . . . 219

dbdMVC Index Sample Controller Class

imageTypesArray() . . . 107

imageType() . . . 107

InputCSS::radio() . . . 25

Generate a radio input.

InputCSS::getXHTML() . . . 24

Generate all markup required for final output.

InputCSS::button() . . . 22

Generate a submit button.

InputCSS::$input_ids . . . 22

Static array of uses input ids.

InputCSS::$div_id . . . 22

Static value to increment to keep track of divs.

InputCSS . . . 21

The InputCSS Class is to generate the markup and JQuery calls to make a fancy CSS form input.

InputCSS::checkbox() . . . 23

Generate a checkbox input.

InputCSS::file() . . . 23

Generate a file/browse input.

InputCSS::getInputID() . . . 24

Generate id for input tag.

InputCSS::getDivID() . . . 24

Generate id for div tag.

InputCSS::getCallingFile() . . . 24

Get the file name of the calling file.

InputCSS.php . . . 4

InputCSS.php :: The Fancy XHTML/CSS Form Input Class

J

JSMin::isAlphaNum() . . . 78

JSMin::get() . . . 78

JSMin::action() . . . 78

JSMin::min() . . . 78

JSMin::minify() . . . 79

JSMinException . . . 79

JSMin::peek() . . . 79

JSMin::next() . . . 79

JSMin::$output . . . 77

JSMin::$lookAhead . . . 77

JSMin::$a . . . 76

JSMin::ORD_SPACE . . . 76

JSMin::ORD_LF . . . 76

JSMin::$b . . . 76

JSMin::$input . . . 76

JSMin::$inputLength . . . 77

JSMin::$inputIndex . . . 77

JSMin . . . 75

jsmin.php - PHP implementation of Douglas Crockford's JSMin.

O

outputAndDestroy() . . . 107

Q

QTLoader::getMovie() . . . 31
Gets (embbeds) quicktime movie in (X)HTML document.

QTLoader::getMovies() . . . 31
Gets movies in list created by addMovie()

QTLoader::getCommonJSPath() . . . 31
Prints include to JavaScript library.

QTLoader::getCommonJS() . . . 30
Prints include to JavaScript library.

QTLoader::getAtts() . . . 30
Parses attribute/value string pairs into an array and returns them combined with current attribute list.

QTLoader::hasMovies() . . . 31
Check to see if there are any movies to load.

QTLoader::loadMovie() . . . 32
Loads (embbeds) quicktime movie in (X)HTML document.

QTLoader::setAtts() . . . 33
Fill the attribute list with the arguments (if any).

QTLoader::optionalAtt() . . . 33
Checks if attribute is optional.

QTLoader::missingReqAtt() . . . 32
Checks attribute list against required attribute list.

QTLoader::loadMovies() . . . 32
Loads movies in list created by addMovie()

QTLoader::buildQTObject() . . . 30
Builds the QTObject JavaScript calls.

QTLoader::addMovie() . . . 29
Adds movie arguments to list of movies to be embedded later by loadMovies().

QTLoader::$codebase . . . 27
codebase without version number

QTLoader::$class_id . . . 27
the quicktime plugin class id

QTLoader::$atts . . . 27
list of main attribute defaults

QTLoader . . . 26
The QTLoader class is used to effortlessly embedd QuickTime movies in a (X)HTML Document.

QTLoader::$common_js_shown . . . 27
flag to be set so we write JavaScript only once

QTLoader::$js_path . . . 28
path to SWFObject external JavaScripts library

QTLoader::$req_atts . . . 29
list of required attribute defaults

QTLoader::$pluginspage . . . 28
plugins page url

QTLoader::$onload . . . 28

QTLoader::$movies . . . 28

list of movies added my addMovie()

QTLoader.php . . . 5

QTLoader.php :: QuickTime movie loading/embedding class

R

resizeImage() . . . 108

reflectImage() . . . 107

S

stretchCopyButton() . . . 108

W

wavEdit::$arrData . . . 220

wavEdit::$bdColor . . . 220

wavEdit::$afiles . . . 220

wavEdit . . . 220

wmString::wkt2kml() . . . 51

Convert WKT to KML.

wmGD.php . . . 106

wavEdit::$bgColor . . . 220

wavEdit::$Db . . . 220

wavEdit::$imgBorder . . . 220

wavEdit::$imgHeight . . . 220

wavEdit::$header . . . 220

wavEdit::$grpColor . . . 220

wavEdit::$gridColor . . . 220

wmString::underLower2SuperCap() . . . 50

Takes an underscored lower case string (under_lower_string) converts it to SuperCaps

wmString::truncate() . . . 50

Truncate a string...

wmString::makePlural() . . . 47

Takes a string and converts to the plural version.

wmString::makeSingular() . . . 47

Takes a string and converts to the singular version.

wmString::lcFirst() . . . 46

Turns the first characher of a string to lower case

wmString::jsSpecialChars() . . . 46

Escapes special characters for use in javascript function calls.

wmString::emailValid() . . . 45

Test if an email address is valid.

wmString::hanukkahStart() . . . 46

Get the unix timestamp of Hanukkah for this year.

wmString::numberSuffix() . . . 47

Add suffixes to numbers.

wmString::randomWord() . . . 48

Generate a random readable word.

wmString::timeLengthFormat() . . . 49

Format a length of time.

wmString::timePastFormat() 50

Format the abount of time past.

wmString::timeFormat() 49

Format the amount of time in seconds to hh::mm:ss

wmString::superCapAddSpaces() 48

Takes a super capped string (SuperCapString) and adds spaces

wmString::superCap2UnderLower() 48

Takes a super capped string (SuperCapString) and adds underscores and makes lowercase

wavEdit::$imgWidth 221

wavEdit::$visual_graph_mode 221

wavEdit::joinFiles() 223

wavEdit::setEcho() 223

wavEdit::getStructure() 223

wavEdit::getHeader() 223

wavEdit::getFileName() 223

wavEdit::getFiles() 223

wavEdit::setFade() 223

wavEdit::setVolume() 224

wavEdit::viewImage() 224

wavEdit::writeWavFile() 224

wavEdit::viewHeader() 224

wavEdit::unpackHeader() 224

wavEdit::unpackData() 224

wavEdit::getDataArr() 222

wavEdit::getBytes() 222

wavEdit::doCrop() 221

wavEdit::doDrawImg() 221

wavEdit::doClip() 221

wavEdit::cropFiles() 221

wavEdit::$wavs 221

wavEdit::doEcho() 221

wavEdit::doFade() 222

wavEdit::doViewHeader() 222

wavEdit::drawImgWave() 222

wavEdit::doSetVol16Bit() 222

wavEdit::doSetVol8Bit() 222

wavEdit::doJoin() 222

wmString::emailEncode() 45

Takes a string (mailto:email@domain.com) and converts it to unicode
to protect it from spiders.

wmString::dayOfWeek() 44

Give the day of week given day number 0-6.

wmSort::multiSort() 39

Sorts a two-dimentional array by one to many key "indexes".

wmSort::_getVal() 40

Finds value for comparison.

wmSort::multiASort() 39

Sorts a two-dimentional array by one to many key "indexes"
and maintains its associative indexes.

wmSort::$sortby 39

wmFileSystem::scanDir() 38

Reads directory contents recursively and returns *them as an associative array.*

wmSort . . . 38

The wmSort class is used for advanced two-dimentional array sorting.

wmSort::_wmMultiSort() . . . 40

Callback function passed to usort() and uasort().

wmString . . . 41

The wmString class is used to manipulate strings in various ways.

wmString::TIME_LEN_LABEL_DAY . . . 41

wmString::TIME_LEN_LABEL_HRS . . . 41

wmString::TIME_LEN_LABEL_AGO . . . 41

wmString::TIME_LEN_FORMAT_SHORT . . . 41

wmString::TIME_LEN_FORMAT_LONG . . . 41

wmFileSystem::mimeType() . . . 37

Get the mime type from mime_magic

wmFileSystem::dirSize() . . . 37

Get the file sixe of a directory and its subcontents in bytes

wmArrays::arrayMergeRecursiveSumFields() . . . 34

Merges multidimentional arrays, and sums the integer values of fields in given list.

wmArrays::arrayRemoveValue() . . . 34

wmArrays . . . 33

The wmArrays class is used to preform complex routines on arrays.

wmString.php . . . 9

wmString.php :: Will Mason's string manipulation class

wmFileSystem.php . . . 7

wmFileSystem.php :: Will Mason's file system class

wmSort.php . . . 8

wmSort.php :: Will Mason's array sorting class

wmArrays::arraySearchRecursive() . . . 35

Searches through a multidimensional array for a given value and if found, returns an array of keys to its location.

wmArrays::arrayValuesRecursive() . . . 35

wmFileSystem . . . 36

The wmFileSystem class is collection of specialized file system utility methods.

wmFileSystem::bytesToHuman() . . . 37

Makes byte counts human readble (i.e. 1024 -> 1KB).

wmArrays::pregExplode() . . . 36

wmArrays::implodeObjArray() . . . 36

wmArrays::explodeNewline() . . . 35

wmString::TIME_LEN_LABEL_LAST . . . 41

wmString::TIME_LEN_LABEL_MIN . . . 42

wmString::TIME_LEN_TYPE_WEEK . . . 43

wmString::TIME_LEN_TYPE_YEAR . . . 43

wmString::TIME_LEN_TYPE_SEC . . . 43

wmString::TIME_LEN_TYPE_MON . . . 43

wmString::TIME_LEN_TYPE_HRS . . . 43

wmString::TIME_LEN_TYPE_MIN . . . 43

wmString::VALID_DATE . . . 43

wmString::VALID_EMAIL . . . 43

wmString::$random_word_vowl . . . 44

wmString::dateValid() . . . 44

Test if a date is valid.

wmString::$random_word_cons 44
wmString::$plural_exceptions 43
wmString::$number_suffixes 43
wmString::TIME_LEN_TYPE_DAY 42
wmString::TIME_LEN_TIME_YEAR 42
wmString::TIME_LEN_LABEL_WEEK 42
wmString::TIME_LEN_LABEL_YEAR 42
wmString::TIME_LEN_LABEL_TODAY 42
wmString::TIME_LEN_LABEL_SEC 42
wmString::TIME_LEN_LABEL_MON 42
wmString::TIME_LEN_LABEL_YESTERDAY 42
wmString::TIME_LEN_TIME_DAY 42
wmString::TIME_LEN_TIME_SEC 42
wmString::TIME_LEN_TIME_WEEK 42
wmString::TIME_LEN_TIME_MON 42
wmString::TIME_LEN_TIME_MIN 42
wmString::TIME_LEN_TIME_HRS 42
wmArrays.php 6
wmArrays.php :: Will Mason's array processing class

_

__autoload() 98
Autoload classes using dbdLoader::loadClass();