

# PHPRum Documentation



# Contents

<a href="#">index.php</a>	0
<a href="#">appconfiguration.class.php</a>	0
<a href="#">applicationbase.class.php</a>	0
<a href="#">appmessage.class.php</a>	0
<a href="#">appmessagetype.class.php</a>	0
<a href="#">appstate.class.php</a>	0
<a href="#">argumentoutofrangeexception.class.php</a>	0
<a href="#">badmembercallexception.class.php</a>	0
<a href="#">badmethodcallexception.class.php</a>	0
<a href="#">classloader.inc.php</a>	0
Function <a href="#">class_search</a>	0
<a href="#">debug.inc.php</a>	0
Function <a href="#">alloc</a>	0
Function <a href="#">breakpoint</a>	0
Function <a href="#">callstack</a>	0
Function <a href="#">clr</a>	0
Function <a href="#">dmp</a>	0
Function <a href="#">stp</a>	0
Function <a href="#">strt</a>	0
Function <a href="#">trace</a>	0
Function <a href="#">trap</a>	0
<a href="#">eventbase.class.php</a>	0
<a href="#">eventhandlerbase.class.php</a>	0
<a href="#">exceptionbase.class.php</a>	0
<a href="#">indexoutofrangeexception.class.php</a>	0
<a href="#">invalidargumentexception.class.php</a>	0
<a href="#">invalidoperationexception.class.php</a>	0
<a href="#">methodnotimplementedexception.class.php</a>	0
<a href="#">missingargumentexception.class.php</a>	0
<a href="#">modelbase.class.php</a>	0
<a href="#">rum.php</a>	0
Define <a href="#">ACTIVERECORD AUTO MAP</a>	0
Define <a href="#">APP CONF FILENAME</a>	0
Define <a href="#">APP PATH</a>	0
Define <a href="#">APP URI</a>	0
Define <a href="#">ASSETS URI</a>	0
Define <a href="#">ASYNC REQUEST PARAMETER</a>	0
Define <a href="#">BUILD PATH</a>	0
Define <a href="#">CACHE PATH</a>	0
Define <a href="#">CACHE TABLENAME</a>	0
Define <a href="#">CLASS EXTENSION</a>	0
Define <a href="#">CONFIG PATH</a>	0
Define <a href="#">CONTROLLERS NAMESPACE</a>	0
Define <a href="#">CONTROLLERS PATH</a>	0

<a href="#">Define</a>	<a href="#">CONTROLLER EXTENSION</a>	0
<a href="#">Define</a>	<a href="#">CONTROLLER TESTCASE SUFFIX</a>	0
<a href="#">Define</a>	<a href="#">DB SCHEMA VERSION TABLENAME</a>	0
<a href="#">Define</a>	<a href="#">DEFAULT TIMEZONE</a>	0
<a href="#">Define</a>	<a href="#">DEPLOYMENT EXTENSION</a>	0
<a href="#">Define</a>	<a href="#">DEPLOY PATH</a>	0
<a href="#">Define</a>	<a href="#">DEV ENV</a>	0
<a href="#">Define</a>	<a href="#">ENV CONF FILENAME</a>	0
<a href="#">Define</a>	<a href="#">ENV PARAMETER</a>	0
<a href="#">Define</a>	<a href="#">ENV PATH</a>	0
<a href="#">Define</a>	<a href="#">ERROR LIMIT</a>	0
<a href="#">Define</a>	<a href="#">FIXTURES PATH</a>	0
<a href="#">Define</a>	<a href="#">FUNCTIONAL TESTS PATH</a>	0
<a href="#">Define</a>	<a href="#">HOST</a>	0
<a href="#">Define</a>	<a href="#">HTDOCS PATH</a>	0
<a href="#">Define</a>	<a href="#">LANGS FILE</a>	0
<a href="#">Define</a>	<a href="#">LANGS TABLENAME</a>	0
<a href="#">Define</a>	<a href="#">LIB PATH</a>	0
<a href="#">Define</a>	<a href="#">LOGS TABLENAME</a>	0
<a href="#">Define</a>	<a href="#">LOG PATH</a>	0
<a href="#">Define</a>	<a href="#">MIGRATIONS PATH</a>	0
<a href="#">Define</a>	<a href="#">MODELS NAMESPACE</a>	0
<a href="#">Define</a>	<a href="#">MODELS PATH</a>	0
<a href="#">Define</a>	<a href="#">NAME</a>	0
<a href="#">Define</a>	<a href="#">PAGE EXTENSION</a>	0
<a href="#">Define</a>	<a href="#">PAGE REQUEST PARAMETER</a>	0
<a href="#">Define</a>	<a href="#">PLUGINS PATH</a>	0
<a href="#">Define</a>	<a href="#">PORT</a>	0
<a href="#">Define</a>	<a href="#">PROTOCOL</a>	0
<a href="#">Define</a>	<a href="#">QUOTE STYLE</a>	0
<a href="#">Define</a>	<a href="#">ROOT</a>	0
<a href="#">Define</a>	<a href="#">SSL PORT</a>	0
<a href="#">Define</a>	<a href="#">SYSTEM PATH</a>	0
<a href="#">Define</a>	<a href="#">TEMPLATES PATH</a>	0
<a href="#">Define</a>	<a href="#">TEMPLATE EXTENSION</a>	0
<a href="#">Define</a>	<a href="#">TESTCASE SUFFIX</a>	0
<a href="#">Define</a>	<a href="#">TEST ENV</a>	0
<a href="#">Define</a>	<a href="#">UNIT TESTS PATH</a>	0
<a href="#">Define</a>	<a href="#">VALIDATION TIMEOUT</a>	0
<a href="#">Define</a>	<a href="#">VIEWS PATH</a>	0
<a href="#">typemismatchexception.class.php</a>		0
<a href="#">consoleapplicationbase.class.php</a>		0
<a href="#">loggerbase.class.php</a>		0
<a href="#">make.class.php</a>		0
<a href="#">rum.class.php</a>		0
<a href="#">tester.class.php</a>		0
<a href="#">gridviewchangeorderevent.class.php</a>		0
<a href="#">gridviewchangeordereventhandler.class.php</a>		0
<a href="#">gridviewfilterevent.class.php</a>		0

<a href="#">gridviewfiltereventhandler.class.php</a>	0
<a href="#">gridviewsortevent.class.php</a>	0
<a href="#">gridviewsorteventhandler.class.php</a>	0
<a href="#">webapplicationbase.class.php</a>	0
<a href="#">Class AppConfiguration</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method loadAppConfig</a>	0
<a href="#">Class ApplicationBase</a>	0
<a href="#">Var \$events</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method execute</a>	0
<a href="#">Method getCache</a>	0
<a href="#">Method getInstance</a>	0
<a href="#">Method getLogger</a>	0
<a href="#">Method getMailClient</a>	0
<a href="#">Method getPluginHooks</a>	0
<a href="#">Method getTranslator</a>	0
<a href="#">Method handleError</a>	0
<a href="#">Method handleException</a>	0
<a href="#">Method loadAppConfig</a>	0
<a href="#">Method run</a>	0
<a href="#">Class AppMessage</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class AppMessageType</a>	0
<a href="#">Method Fail</a>	0
<a href="#">Method Info</a>	0
<a href="#">Method Success</a>	0
<a href="#">Method Warning</a>	0
<a href="#">Method __toString</a>	0
<a href="#">Class AppState</a>	0
<a href="#">Method Debug</a>	0
<a href="#">Method On</a>	0
<a href="#">Class ArgumentOutOfRangeException</a>	0
<a href="#">Class BadMemberCallException</a>	0
<a href="#">Class BadMethodCallException</a>	0
<a href="#">Class ConsoleApplicationBase</a>	0
<a href="#">Class EventBase</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class EventHandlerBase</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method raise</a>	0
<a href="#">Class ExceptionBase</a>	0
<a href="#">Class GridViewChangeOrderEvent</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class GridViewChangeOrderEventHandler</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class GridViewFilterEvent</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class GridViewFilterEventHandler</a>	0

<a href="#">Constructor</a> <a href="#">construct</a>	0
<a href="#">Class GridViewSortEvent</a>	0
<a href="#">Constructor</a> <a href="#">construct</a>	0
<a href="#">Class GridViewSortEventHandler</a>	0
<a href="#">Constructor</a> <a href="#">construct</a>	0
<a href="#">Class IndexOutOfRangeException</a>	0
<a href="#">Class InvalidArgumentException</a>	0
<a href="#">Class InvalidOperationException</a>	0
<a href="#">Class LoggerBase</a>	0
<a href="#">Method</a> <a href="#">flush</a>	0
<a href="#">Method</a> <a href="#">log</a>	0
<a href="#">Method</a> <a href="#">logs</a>	0
<a href="#">Class Make</a>	0
<a href="#">Var</a> <a href="#">\$namespace</a>	0
<a href="#">Method</a> <a href="#">execute</a>	0
<a href="#">Method</a> <a href="#">getOptions</a>	0
<a href="#">Method</a> <a href="#">handleError</a>	0
<a href="#">Method</a> <a href="#">handleException</a>	0
<a href="#">Class MethodNotImplementedException</a>	0
<a href="#">Var</a> <a href="#">\$message</a>	0
<a href="#">Class MissingArgumentException</a>	0
<a href="#">Class ModelBase</a>	0
<a href="#">Method</a> <a href="#">getClass</a>	0
<a href="#">Method</a> <a href="#">toArray</a>	0
<a href="#">Class Rum</a>	0
<a href="#">Method</a> <a href="#">app</a>	0
<a href="#">Method</a> <a href="#">baseURI</a>	0
<a href="#">Method</a> <a href="#">breadcrumb</a>	0
<a href="#">Method</a> <a href="#">cache</a>	0
<a href="#">Method</a> <a href="#">clearMessages</a>	0
<a href="#">Method</a> <a href="#">config</a>	0
<a href="#">Method</a> <a href="#">db</a>	0
<a href="#">Method</a> <a href="#">escape</a>	0
<a href="#">Method</a> <a href="#">flash</a>	0
<a href="#">Method</a> <a href="#">forward</a>	0
<a href="#">Method</a> <a href="#">link</a>	0
<a href="#">Method</a> <a href="#">log</a>	0
<a href="#">Method</a> <a href="#">messages</a>	0
<a href="#">Method</a> <a href="#">requestHandler</a>	0
<a href="#">Method</a> <a href="#">sendHTTPError</a>	0
<a href="#">Method</a> <a href="#">tl</a>	0
<a href="#">Method</a> <a href="#">trace</a>	0
<a href="#">Method</a> <a href="#">uri</a>	0
<a href="#">Method</a> <a href="#">url</a>	0
<a href="#">Class Tester</a>	0
<a href="#">Method</a> <a href="#">getAllFunctionalTestCases</a>	0
<a href="#">Method</a> <a href="#">getAllUnitTestCases</a>	0
<a href="#">Method</a> <a href="#">getFunctionalTestCase</a>	0
<a href="#">Method</a> <a href="#">getUnitTestCase</a>	0

<a href="#">Method loadTestConfig</a>	0
<a href="#">Method runAllTestCases</a>	0
<a href="#">Method runFunctionalTestCase</a>	0
<a href="#">Method runUnitTestCase</a>	0
<a href="#">Class TypeMismatchException</a>	0
<a href="#">Class WebApplicationBase</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method clearForwardPage</a>	0
<a href="#">Method execute</a>	0
<a href="#">Method getAllControllers</a>	0
<a href="#">Method getPageURI</a>	0
<a href="#">Method getRequestHandler</a>	0
<a href="#">Method getSession</a>	0
<a href="#">Method handleError</a>	0
<a href="#">Method handleException</a>	0
<a href="#">Method loadApplicationState</a>	0
<a href="#">Method requestProcessor</a>	0
<a href="#">Method saveApplicationState</a>	0
<a href="#">Method sendHTTPError</a>	0
<a href="#">Method setForwardPage</a>	0
<a href="#">Method trace</a>	0
<a href="#">activerecordbase.class.php</a>	0
<a href="#">activerecordcollection.class.php</a>	0
<a href="#">activerecordexception.class.php</a>	0
<a href="#">relationshipstype.class.php</a>	0
<a href="#">Class ActiveRecordBase</a>	0
<a href="#">Var \$pkey</a>	0
<a href="#">Var \$relationships</a>	0
<a href="#">Var \$table</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method afterCreate</a>	0
<a href="#">Method afterDelete</a>	0
<a href="#">Method afterInsert</a>	0
<a href="#">Method afterRetrieve</a>	0
<a href="#">Method afterSave</a>	0
<a href="#">Method afterUpdate</a>	0
<a href="#">Method all</a>	0
<a href="#">Method beforeCreate</a>	0
<a href="#">Method beforeDelete</a>	0
<a href="#">Method beforeInsert</a>	0
<a href="#">Method beforeRetrieve</a>	0
<a href="#">Method beforeSave</a>	0
<a href="#">Method beforeUpdate</a>	0
<a href="#">Method count</a>	0
<a href="#">Method create</a>	0
<a href="#">Method delete</a>	0
<a href="#">Method deleteAll</a>	0
<a href="#">Method find</a>	0
<a href="#">Method findAll</a>	0

<a href="#">Method findById</a>	0
<a href="#">Method first</a>	0
<a href="#">Method form</a>	0
<a href="#">Method last</a>	0
<a href="#">Method more</a>	0
<a href="#">Method save</a>	0
<a href="#">Method toArray</a>	0
<a href="#">Class ActiveRecordCollection</a>	0
<a href="#">Method add</a>	0
<a href="#">Method itemAt</a>	0
<a href="#">Class ActiveRecordException</a>	0
<a href="#">Class RelationshipType</a>	0
<a href="#">Method BelongsTo</a>	0
<a href="#">Method HasMany</a>	0
<a href="#">Method HasManyAndBelongsTo</a>	0
<a href="#">Method toString</a>	0
<a href="#">applicationrunevent.class.php</a>	0
<a href="#">applicationruneventhandler.class.php</a>	0
<a href="#">pagecontrollercreatepageevent.class.php</a>	0
<a href="#">pagecontrollercreatepageeventhandler.class.php</a>	0
<a href="#">webapplicationhandlerequestevent.class.php</a>	0
<a href="#">webapplicationhandlerequesteventhandler.class.php</a>	0
<a href="#">webapplicationloadstateevent.class.php</a>	0
<a href="#">webapplicationloadstateeventhandler.class.php</a>	0
<a href="#">webapplicationsavestateevent.class.php</a>	0
<a href="#">webapplicationsavestateeventhandler.class.php</a>	0
<a href="#">Class ApplicationRunEvent</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class ApplicationRunEventHandler</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class PageControllerCreatePageEvent</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class PageControllerCreatePageEventHandler</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class WebApplicationHandleRequestEvent</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class WebApplicationHandleRequestEventHandler</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class WebApplicationLoadStateEvent</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class WebApplicationLoadStateEventHandler</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class WebApplicationSaveStateEvent</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class WebApplicationSaveStateEventHandler</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">build.class.php</a>	0
<a href="#">cachebase.class.php</a>	0
<a href="#">cacheexception.class.php</a>	0

<a href="#">dbcache.class.php</a>	0
<a href="#">filecache.class.php</a>	0
<a href="#">memcache.class.php</a>	0
<a href="#">Class Build</a>	0
<a href="#">Var \$verbose</a>	0
<a href="#">Method build</a>	0
<a href="#">Method clean</a>	0
<a href="#">Method clear</a>	0
<a href="#">Method get</a>	0
<a href="#">Method getFileArray</a>	0
<a href="#">Method put</a>	0
<a href="#">Class CacheBase</a>	0
<a href="#">Method clear</a>	0
<a href="#">Method flush</a>	0
<a href="#">Method get</a>	0
<a href="#">Method put</a>	0
<a href="#">Class CacheException</a>	0
<a href="#">Class DBCache</a>	0
<a href="#">Constructor    construct</a>	0
<a href="#">Method clear</a>	0
<a href="#">Method flush</a>	0
<a href="#">Method get</a>	0
<a href="#">Method getIdArray</a>	0
<a href="#">Method put</a>	0
<a href="#">Class FileCache</a>	0
<a href="#">Method clear</a>	0
<a href="#">Method flush</a>	0
<a href="#">Method get</a>	0
<a href="#">Method getIdArray</a>	0
<a href="#">Method put</a>	0
<a href="#">Class MEMCache</a>	0
<a href="#">Constructor    construct</a>	0
<a href="#">Method clear</a>	0
<a href="#">Method flush</a>	0
<a href="#">Method get</a>	0
<a href="#">Method put</a>	0
<a href="#">appmessagecollection.class.php</a>	0
<a href="#">appsettingscollection.class.php</a>	0
<a href="#">eventargs.class.php</a>	0
<a href="#">eventcollection.class.php</a>	0
<a href="#">eventhandlercollection.class.php</a>	0
<a href="#">collectionbase.class.php</a>	0
<a href="#">dictionarybase.class.php</a>	0
<a href="#">iteratorbase.class.php</a>	0
<a href="#">keycollection.class.php</a>	0
<a href="#">stringcollection.class.php</a>	0
<a href="#">stringdictionary.class.php</a>	0
<a href="#">valuecollection.class.php</a>	0
<a href="#">Class AppMessageCollection</a>	0



<a href="#">Method add</a>	0
<a href="#">Method getByType</a>	0
<a href="#">Method itemAt</a>	0
<a href="#">Method toString</a>	0
<a href="#">Class AppSettingsCollection</a>	0
<a href="#">Class CollectionBase</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method add</a>	0
<a href="#">Method contains</a>	0
<a href="#">Method indexOf</a>	0
<a href="#">Method itemAt</a>	0
<a href="#">Method remove</a>	0
<a href="#">Method removeAll</a>	0
<a href="#">Method removeAt</a>	0
<a href="#">Class DictionaryBase</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method add</a>	0
<a href="#">Method contains</a>	0
<a href="#">Method indexOf</a>	0
<a href="#">Method item</a>	0
<a href="#">Method itemAt</a>	0
<a href="#">Method keyAt</a>	0
<a href="#">Method remove</a>	0
<a href="#">Method removeAll</a>	0
<a href="#">Method removeAt</a>	0
<a href="#">Class EventArgs</a>	0
<a href="#">Class EventCollection</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method contains</a>	0
<a href="#">Method indexOf</a>	0
<a href="#">Method raise</a>	0
<a href="#">Method registerEventHandler</a>	0
<a href="#">Class EventHandlerCollection</a>	0
<a href="#">Method raise</a>	0
<a href="#">Method registerEventHandler</a>	0
<a href="#">Class IteratorBase</a>	0
<a href="#">Var \$items</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method toArray</a>	0
<a href="#">Class KeyCollection</a>	0
<a href="#">Method itemAt</a>	0
<a href="#">Class StringCollection</a>	0
<a href="#">Method add</a>	0
<a href="#">Method itemAt</a>	0
<a href="#">Method trim</a>	0
<a href="#">Class StringDictionary</a>	0
<a href="#">Method add</a>	0
<a href="#">Method item</a>	0
<a href="#">Method itemAt</a>	0

<a href="#">Method keyAt</a>	0
<a href="#">Method trim</a>	0
Class <a href="#">ValueCollection</a>	0
<a href="#">Method itemAt</a>	0
<a href="#">httpwebrequest.class.php</a>	0
<a href="#">httpwebresponse.class.php</a>	0
Class <a href="#">HTTPWebRequest</a>	0
<a href="#">Method addHeader</a>	0
<a href="#">Method getRequestStream</a>	0
<a href="#">Method getRequestString</a>	0
<a href="#">Method getResponse</a>	0
<a href="#">Method sendRequest</a>	0
<a href="#">Method setData</a>	0
Class <a href="#">HTTPWebResponse</a>	0
<a href="#">Method readFromStream</a>	0
<a href="#">columncompare.class.php</a>	0
<a href="#">columnschema.class.php</a>	0
<a href="#">dataadapter.class.php</a>	0
<a href="#">dataadapterexception.class.php</a>	0
<a href="#">databaseexception.class.php</a>	0
<a href="#">databaseschema.class.php</a>	0
<a href="#">dataset.class.php</a>	0
<a href="#">datasettype.class.php</a>	0
<a href="#">dirdataadapter.class.php</a>	0
<a href="#">mssqldataadapter.class.php</a>	0
<a href="#">mssqlquerybuilder.class.php</a>	0
<a href="#">mssqltransaction.class.php</a>	0
<a href="#">mysqldataadapter.class.php</a>	0
<a href="#">mysqlquerybuilder.class.php</a>	0
<a href="#">mysqltransaction.class.php</a>	0
<a href="#">mysqldataadapter.class.php</a>	0
<a href="#">mysqlquerybuilder.class.php</a>	0
<a href="#">mysqltransaction.class.php</a>	0
<a href="#">pdodataadapter.class.php</a>	0
<a href="#">pdoquerybuilder.class.php</a>	0
<a href="#">pdotransaction.class.php</a>	0
<a href="#">querybuilderbase.class.php</a>	0
<a href="#">queryexception.class.php</a>	0
<a href="#">schemaexception.class.php</a>	0
<a href="#">tableschema.class.php</a>	0
<a href="#">textdataadapter.class.php</a>	0
<a href="#">transactionbase.class.php</a>	0
<a href="#">transactionexception.class.php</a>	0
Class <a href="#">ColumnCompare</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method compareDateString</a>	0
<a href="#">Method compareInt</a>	0
<a href="#">Method compareNumeric</a>	0
<a href="#">Method compareString</a>	0

<a href="#">Method compareStringi</a>	0
<a href="#">Class ColumnSchema</a>	0
<a href="#">Constructor   construct</a>	0
<a href="#">Method     get</a>	0
<a href="#">Class DataAdapter</a>	0
<a href="#">Var \$args</a>	0
<a href="#">Var \$cacheEnabled</a>	0
<a href="#">Var \$cacheExpires</a>	0
<a href="#">Var \$charset</a>	0
<a href="#">Var \$collation</a>	0
<a href="#">Constructor   construct</a>	0
<a href="#">Method addTableSchema</a>	0
<a href="#">Method alterTableSchema</a>	0
<a href="#">Method beginTransaction</a>	0
<a href="#">Method buildSchema</a>	0
<a href="#">Method close</a>	0
<a href="#">Method create</a>	0
<a href="#">Method delete</a>	0
<a href="#">Method disableCaching</a>	0
<a href="#">Method dropTableSchema</a>	0
<a href="#">Method enableCaching</a>	0
<a href="#">Method escapeString</a>	0
<a href="#">Method execute</a>	0
<a href="#">Method executeBatch</a>	0
<a href="#">Method fill</a>	0
<a href="#">Method getAffectedRows</a>	0
<a href="#">Method getLastInsertId</a>	0
<a href="#">Method getLastQuery</a>	0
<a href="#">Method getLastQueryTime</a>	0
<a href="#">Method getQueryCount</a>	0
<a href="#">Method getQueryTime</a>	0
<a href="#">Method getSchema</a>	0
<a href="#">Method insert</a>	0
<a href="#">Method open</a>	0
<a href="#">Method openDataSet</a>	0
<a href="#">Method opened</a>	0
<a href="#">Method query</a>	0
<a href="#">Method queryBuilder</a>	0
<a href="#">Method rebuildSchema</a>	0
<a href="#">Method runQuery</a>	0
<a href="#">Method update</a>	0
<a href="#">Class DataAdapterException</a>	0
<a href="#">Class DatabaseException</a>	0
<a href="#">Class DatabaseSchema</a>	0
<a href="#">Constructor   construct</a>	0
<a href="#">Method seek</a>	0
<a href="#">Method     get</a>	0
<a href="#">Class DataSet</a>	0
<a href="#">Method addNew</a>	0

<a href="#">Method bof</a>	0
<a href="#">Method delete</a>	0
<a href="#">Method eof</a>	0
<a href="#">Method filter</a>	0
<a href="#">Method first</a>	0
<a href="#">Method getAvg</a>	0
<a href="#">Method getCSVString</a>	0
<a href="#">Method getMax</a>	0
<a href="#">Method getMin</a>	0
<a href="#">Method getSum</a>	0
<a href="#">Method getXMLString</a>	0
<a href="#">Method insert</a>	0
<a href="#">Method last</a>	0
<a href="#">Method move</a>	0
<a href="#">Method next</a>	0
<a href="#">Method page</a>	0
<a href="#">Method pageCount</a>	0
<a href="#">Method prev</a>	0
<a href="#">Method requery</a>	0
<a href="#">Method seek</a>	0
<a href="#">Method setFieldMeta</a>	0
<a href="#">Method setFields</a>	0
<a href="#">Method setRows</a>	0
<a href="#">Method setTable</a>	0
<a href="#">Method sort</a>	0
<a href="#">Method toArray</a>	0
<a href="#">Method update</a>	0
<a href="#">Method updateRowData</a>	0
<a href="#">Method __get</a>	0
<a href="#">Class DataSetType</a>	0
<a href="#">Method OpenDynamic</a>	0
<a href="#">Method OpenReadonly</a>	0
<a href="#">Method OpenStatic</a>	0
<a href="#">Class DirDataAdapter</a>	0
<a href="#">Method addTableSchema</a>	0
<a href="#">Method alterTableSchema</a>	0
<a href="#">Method beginTransaction</a>	0
<a href="#">Method buildSchema</a>	0
<a href="#">Method close</a>	0
<a href="#">Method delete</a>	0
<a href="#">Method dropTableSchema</a>	0
<a href="#">Method escapeString</a>	0
<a href="#">Method fill</a>	0
<a href="#">Method getAffectedRows</a>	0
<a href="#">Method getLastInsertId</a>	0
<a href="#">Method insert</a>	0
<a href="#">Method open</a>	0
<a href="#">Method opened</a>	0
<a href="#">Method query</a>	0

<a href="#">Method queryBuilder</a>	0
<a href="#">Method update</a>	0
<a href="#">Class MSSQLDataAdapter</a>	0
<a href="#">Var \$charset</a>	0
<a href="#">Method addTableSchema</a>	0
<a href="#">Method alterTableSchema</a>	0
<a href="#">Method begin</a>	0
<a href="#">Method beginTransaction</a>	0
<a href="#">Method buildSchema</a>	0
<a href="#">Method close</a>	0
<a href="#">Method commit</a>	0
<a href="#">Method delete</a>	0
<a href="#">Method dropTableSchema</a>	0
<a href="#">Method escapeString</a>	0
<a href="#">Method executeInternal</a>	0
<a href="#">Method fill</a>	0
<a href="#">Method getAffectedRows</a>	0
<a href="#">Method getField</a>	0
<a href="#">Method getLastColumnIdOfTable</a>	0
<a href="#">Method getLastInsertId</a>	0
<a href="#">Method getQuery</a>	0
<a href="#">Method getTables</a>	0
<a href="#">Method insert</a>	0
<a href="#">Method open</a>	0
<a href="#">Method opened</a>	0
<a href="#">Method query</a>	0
<a href="#">Method queryBuilder</a>	0
<a href="#">Method rollback</a>	0
<a href="#">Method setForeignKeyChecks</a>	0
<a href="#">Method update</a>	0
<a href="#">Class MSSQLQueryBuilder</a>	0
<a href="#">Method addColumn</a>	0
<a href="#">Method addGroupByClause</a>	0
<a href="#">Method addHavingClause</a>	0
<a href="#">Method addJoin</a>	0
<a href="#">Method addOrderByClause</a>	0
<a href="#">Method addTable</a>	0
<a href="#">Method addValue</a>	0
<a href="#">Method addWhereClause</a>	0
<a href="#">Method getQuery</a>	0
<a href="#">Method is_date</a>	0
<a href="#">Class MSSQLTransaction</a>	0
<a href="#">Method beginTransaction</a>	0
<a href="#">Method commitTransaction</a>	0
<a href="#">Method rollbackTransaction</a>	0
<a href="#">Class MySQLDataAdapter</a>	0
<a href="#">Var \$charset</a>	0
<a href="#">Method addTableSchema</a>	0
<a href="#">Method alterTableSchema</a>	0

<a href="#">Method beginTransaction</a>	0
<a href="#">Method buildSchema</a>	0
<a href="#">Method close</a>	0
<a href="#">Method delete</a>	0
<a href="#">Method dropTableSchema</a>	0
<a href="#">Method escapeString</a>	0
<a href="#">Method fill</a>	0
<a href="#">Method getAffectedRows</a>	0
<a href="#">Method getLastInsertId</a>	0
<a href="#">Method insert</a>	0
<a href="#">Method open</a>	0
<a href="#">Method opened</a>	0
<a href="#">Method query</a>	0
<a href="#">Method queryBuilder</a>	0
<a href="#">Method update</a>	0
<a href="#">Class MySQLiDataAdapter</a>	0
<a href="#">Var \$charset</a>	0
<a href="#">Method addTableSchema</a>	0
<a href="#">Method alterTableSchema</a>	0
<a href="#">Method beginTransaction</a>	0
<a href="#">Method buildSchema</a>	0
<a href="#">Method close</a>	0
<a href="#">Method delete</a>	0
<a href="#">Method dropTableSchema</a>	0
<a href="#">Method escapeString</a>	0
<a href="#">Method fill</a>	0
<a href="#">Method getAffectedRows</a>	0
<a href="#">Method getLastInsertId</a>	0
<a href="#">Method insert</a>	0
<a href="#">Method open</a>	0
<a href="#">Method opened</a>	0
<a href="#">Method query</a>	0
<a href="#">Method queryBuilder</a>	0
<a href="#">Method update</a>	0
<a href="#">Class MySQLiQueryBuilder</a>	0
<a href="#">Method addColumn</a>	0
<a href="#">Method addGroupByClause</a>	0
<a href="#">Method addHavingClause</a>	0
<a href="#">Method addJoin</a>	0
<a href="#">Method addOrderByClause</a>	0
<a href="#">Method addTable</a>	0
<a href="#">Method addValue</a>	0
<a href="#">Method addWhereClause</a>	0
<a href="#">Method getQuery</a>	0
<a href="#">Class MySQLiTransaction</a>	0
<a href="#">Method beginTransaction</a>	0
<a href="#">Method commitTransaction</a>	0
<a href="#">Method rollbackTransaction</a>	0
<a href="#">Class MySQLQueryBuilder</a>	0

<a href="#">Method addColumn</a>	0
<a href="#">Method addGroupByClause</a>	0
<a href="#">Method addHavingClause</a>	0
<a href="#">Method addJoin</a>	0
<a href="#">Method addOrderByClause</a>	0
<a href="#">Method addTable</a>	0
<a href="#">Method addValue</a>	0
<a href="#">Method addWhereClause</a>	0
<a href="#">Method getQuery</a>	0
<a href="#">Class MySQLTransaction</a>	0
<a href="#">Method beginTransaction</a>	0
<a href="#">Method commitTransaction</a>	0
<a href="#">Method rollbackTransaction</a>	0
<a href="#">Class PDODataAdapter</a>	0
<a href="#">Method addTableSchema</a>	0
<a href="#">Method alterTableSchema</a>	0
<a href="#">Method beginTransaction</a>	0
<a href="#">Method buildSchema</a>	0
<a href="#">Method close</a>	0
<a href="#">Method delete</a>	0
<a href="#">Method dropTableSchema</a>	0
<a href="#">Method escapeString</a>	0
<a href="#">Method fill</a>	0
<a href="#">Method getAffectedRows</a>	0
<a href="#">Method getField</a>	0
<a href="#">Method getLastInsertId</a>	0
<a href="#">Method insert</a>	0
<a href="#">Method open</a>	0
<a href="#">Method opened</a>	0
<a href="#">Method query</a>	0
<a href="#">Method queryBuilder</a>	0
<a href="#">Method update</a>	0
<a href="#">Class PDOQueryBuilder</a>	0
<a href="#">Method addColumn</a>	0
<a href="#">Method addGroupByClause</a>	0
<a href="#">Method addHavingClause</a>	0
<a href="#">Method addJoin</a>	0
<a href="#">Method addOrderByClause</a>	0
<a href="#">Method addTable</a>	0
<a href="#">Method addValue</a>	0
<a href="#">Method addWhereClause</a>	0
<a href="#">Method getMSSQLQuery</a>	0
<a href="#">Method getMySQLQuery</a>	0
<a href="#">Method getQuery</a>	0
<a href="#">Method is_date</a>	0
<a href="#">Class PDOTransaction</a>	0
<a href="#">Method beginTransaction</a>	0
<a href="#">Method commitTransaction</a>	0
<a href="#">Method rollbackTransaction</a>	0

<a href="#">Class QueryBuilderInterface</a>	0
<a href="#">Var \$dataAdapter</a>	0
<a href="#">Var \$empty</a>	0
<a href="#">Var \$statement</a>	0
<a href="#">Constructor   construct</a>	0
<a href="#">Method addColumn</a>	0
<a href="#">Method addGroupByClause</a>	0
<a href="#">Method addHavingClause</a>	0
<a href="#">Method addJoin</a>	0
<a href="#">Method addOrderByClause</a>	0
<a href="#">Method addTable</a>	0
<a href="#">Method addValue</a>	0
<a href="#">Method addWhereClause</a>	0
<a href="#">Method delete</a>	0
<a href="#">Method from</a>	0
<a href="#">Method getQuery</a>	0
<a href="#">Method groupBy</a>	0
<a href="#">Method having</a>	0
<a href="#">Method innerJoin</a>	0
<a href="#">Method insertInto</a>	0
<a href="#">Method leftJoin</a>	0
<a href="#">Method openDataSet</a>	0
<a href="#">Method orderBy</a>	0
<a href="#">Method rightJoin</a>	0
<a href="#">Method runQuery</a>	0
<a href="#">Method select</a>	0
<a href="#">Method set</a>	0
<a href="#">Method setColumns</a>	0
<a href="#">Method truncate</a>	0
<a href="#">Method update</a>	0
<a href="#">Method values</a>	0
<a href="#">Method where</a>	0
<a href="#">Class QueryException</a>	0
<a href="#">Class SchemaException</a>	0
<a href="#">Class TableSchema</a>	0
<a href="#">Constructor   construct</a>	0
<a href="#">Method seek</a>	0
<a href="#">Method   __get</a>	0
<a href="#">Class TextDataAdapter</a>	0
<a href="#">Method addTableSchema</a>	0
<a href="#">Method alterTableSchema</a>	0
<a href="#">Method beginTransaction</a>	0
<a href="#">Method buildSchema</a>	0
<a href="#">Method close</a>	0
<a href="#">Method delete</a>	0
<a href="#">Method dropTableSchema</a>	0
<a href="#">Method escapeString</a>	0
<a href="#">Method fill</a>	0
<a href="#">Method getAffectedRows</a>	0



<a href="#">Method getLastInsertId</a>	0
<a href="#">Method insert</a>	0
<a href="#">Method open</a>	0
<a href="#">Method opened</a>	0
<a href="#">Method query</a>	0
<a href="#">Method queryBuilder</a>	0
<a href="#">Method update</a>	0
<a href="#">Class TransactionBase</a>	0
<a href="#">Var \$dataAdapter</a>	0
<a href="#">Constructor   construct</a>	0
<a href="#">Method begin</a>	0
<a href="#">Method beginTransaction</a>	0
<a href="#">Method commit</a>	0
<a href="#">Method commitTransaction</a>	0
<a href="#">Method rollback</a>	0
<a href="#">Method rollbackTransaction</a>	0
<a href="#">Class TransactionException</a>	0
<a href="#">deploy.class.php</a>	0
<a href="#">deploymentbase.class.php</a>	0
<a href="#">sshclientbase.class.php</a>	0
<a href="#">get.class.php</a>	0
<a href="#">Class Deploy</a>	0
<a href="#">Method execute</a>	0
<a href="#">Method getOptions</a>	0
<a href="#">Method handleError</a>	0
<a href="#">Method handleException</a>	0
<a href="#">Class DeploymentBase</a>	0
<a href="#">Var \$home_path</a>	0
<a href="#">Var \$release_path</a>	0
<a href="#">Constructor   construct</a>	0
<a href="#">Class Get</a>	0
<a href="#">Method execute</a>	0
<a href="#">Method getOptions</a>	0
<a href="#">Method handleError</a>	0
<a href="#">Method handleException</a>	0
<a href="#">Class SSHClientBase</a>	0
<a href="#">Var \$password</a>	0
<a href="#">Var \$port</a>	0
<a href="#">Var \$server</a>	0
<a href="#">Var \$username</a>	0
<a href="#">Method exec</a>	0
<a href="#">Method put</a>	0
<a href="#">Method run</a>	0
<a href="#">dbtranslator.class.php</a>	0
<a href="#">filetranslator.class.php</a>	0
<a href="#">translatorbase.class.php</a>	0
<a href="#">translatorexception.class.php</a>	0
<a href="#">Class DBTranslator</a>	0
<a href="#">Constructor   construct</a>	0

<a href="#">Method get</a>	0
<a href="#">Class FileTranslator</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method get</a>	0
<a href="#">Class TranslatorBase</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method get</a>	0
<a href="#">Method setCharset</a>	0
<a href="#">Method setLang</a>	0
<a href="#">Class TranslatorException</a>	0
<a href="#">directorynotfoundexception.class.php</a>	0
<a href="#">directorynotwritableexception.class.php</a>	0
<a href="#">file.class.php</a>	0
<a href="#">fileloadexception.class.php</a>	0
<a href="#">filenotfoundexception.class.php</a>	0
<a href="#">filenotwritableexception.class.php</a>	0
<a href="#">filestream.class.php</a>	0
<a href="#">filesystem.class.php</a>	0
<a href="#">folder.class.php</a>	0
<a href="#">ioexception.class.php</a>	0
<a href="#">socketstream.class.php</a>	0
<a href="#">streambase.class.php</a>	0
<a href="#">httprequest.class.php</a>	0
<a href="#">httpresponse.class.php</a>	0
<a href="#">session.class.php</a>	0
<a href="#">Class DirectoryNotFoundException</a>	0
<a href="#">Class DirectoryNotWritableException</a>	0
<a href="#">Class File</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method changeFileMode</a>	0
<a href="#">Method copy</a>	0
<a href="#">Method delete</a>	0
<a href="#">Method move</a>	0
<a href="#">Method readLastLine</a>	0
<a href="#">Method rename</a>	0
<a href="#">Class FileLoadException</a>	0
<a href="#">Class FileNotFoundException</a>	0
<a href="#">Class FileNotWritableException</a>	0
<a href="#">Class FileStream</a>	0
<a href="#">Method open</a>	0
<a href="#">Class FileSystem</a>	0
<a href="#">Method changeFileMode</a>	0
<a href="#">Method copy</a>	0
<a href="#">Method delete</a>	0
<a href="#">Method move</a>	0
<a href="#">Method removeDirectory</a>	0
<a href="#">Method rename</a>	0
<a href="#">Class Folder</a>	0
<a href="#">Constructor construct</a>	0

<a href="#">Method changeFileMode</a>	0
<a href="#">Method copy</a>	0
<a href="#">Method delete</a>	0
<a href="#">Method move</a>	0
<a href="#">Method rename</a>	0
<a href="#">Class HTTPRequest</a>	0
<a href="#">Var \$cookie</a>	0
<a href="#">Var \$get</a>	0
<a href="#">Var \$post</a>	0
<a href="#">Var \$request</a>	0
<a href="#">Constructor   construct</a>	0
<a href="#">Method getQueryString</a>	0
<a href="#">Method getRawPostData</a>	0
<a href="#">Method getRequestHeaders</a>	0
<a href="#">Method getRequestMethod</a>	0
<a href="#">Method getRequestReferer</a>	0
<a href="#">Method getRequestURI</a>	0
<a href="#">Class HTTPResponse</a>	0
<a href="#">Constructor   construct</a>	0
<a href="#">Method addHeader</a>	0
<a href="#">Method clear</a>	0
<a href="#">Method end</a>	0
<a href="#">Method flush</a>	0
<a href="#">Method getResponseContent</a>	0
<a href="#">Method getResponseHeaders</a>	0
<a href="#">Method redirect</a>	0
<a href="#">Method setCookie</a>	0
<a href="#">Method write</a>	0
<a href="#">Class IOException</a>	0
<a href="#">Class Session</a>	0
<a href="#">Constructor   construct</a>	0
<a href="#">Method close</a>	0
<a href="#">Method destroy</a>	0
<a href="#">Method getSessionData</a>	0
<a href="#">Method read</a>	0
<a href="#">Method regenerateId</a>	0
<a href="#">Method removeAll</a>	0
<a href="#">Method start</a>	0
<a href="#">Method write</a>	0
<a href="#">Class SocketStream</a>	0
<a href="#">Method open</a>	0
<a href="#">Class StreamBase</a>	0
<a href="#">Var \$handle</a>	0
<a href="#">Method close</a>	0
<a href="#">Method eos</a>	0
<a href="#">Method flush</a>	0
<a href="#">Method getPosition</a>	0
<a href="#">Method length</a>	0
<a href="#">Method read</a>	0

<a href="#">Method readln</a>	0
<a href="#">Method setBOS</a>	0
<a href="#">Method setEOS</a>	0
<a href="#">Method setPosition</a>	0
<a href="#">Method write</a>	0
<a href="#">Method writeln</a>	0
<a href="#">dblogger.class.php</a>	0
<a href="#">filelogger.class.php</a>	0
<a href="#">loggerexception.class.php</a>	0
<a href="#">Class DBLogger</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method flush</a>	0
<a href="#">Method log</a>	0
<a href="#">Method logs</a>	0
<a href="#">Class FileLogger</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method flush</a>	0
<a href="#">Method log</a>	0
<a href="#">Method logs</a>	0
<a href="#">Class LoggerException</a>	0
<a href="#">imailclient.class.php</a>	0
<a href="#">mailmessage.class.php</a>	0
<a href="#">phpmailclient.class.php</a>	0
<a href="#">sendmailclient.class.php</a>	0
<a href="#">smtpclient.class.php</a>	0
<a href="#">Class IMailClient</a>	0
<a href="#">Method send</a>	0
<a href="#">Class MailMessage</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method addBcc</a>	0
<a href="#">Method addCc</a>	0
<a href="#">Method attach</a>	0
<a href="#">Method getContent</a>	0
<a href="#">Method getHeaders</a>	0
<a href="#">Method setBody</a>	0
<a href="#">Method setCharset</a>	0
<a href="#">Method setContenttype</a>	0
<a href="#">Method setEncoding</a>	0
<a href="#">Method setFrom</a>	0
<a href="#">Method setSubject</a>	0
<a href="#">Method setTo</a>	0
<a href="#">Class PHPMailClient</a>	0
<a href="#">Method send</a>	0
<a href="#">Class SendMailClient</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method send</a>	0
<a href="#">Class SMTPClient</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method close</a>	0

<a href="#">Method open</a>	0
<a href="#">Method send</a>	0
<a href="#">makebase.class.php</a>	0
<a href="#">controller.php</a>	0
<a href="#">formcontroller.php</a>	0
<a href="#">gridviewController.php</a>	0
<a href="#">jsonwebservice.php</a>	0
<a href="#">listcontroller.php</a>	0
<a href="#">model.php</a>	0
<a href="#">pagecontroller.php</a>	0
<a href="#">restwebservice.php</a>	0
<a href="#">soapwebservice.php</a>	0
<a href="#">Class Controller</a>	0
<a href="#">Class Constant ControllerNamespace</a>	0
<a href="#">Method make</a>	0
<a href="#">Class FormController</a>	0
<a href="#">Class Constant ControllerNamespace</a>	0
<a href="#">Class Constant DefaultBaseController</a>	0
<a href="#">Class Constant ModelNamespace</a>	0
<a href="#">Method make</a>	0
<a href="#">Class GridViewController</a>	0
<a href="#">Class Constant ControllerNamespace</a>	0
<a href="#">Class Constant DefaultBaseController</a>	0
<a href="#">Class Constant ModelNamespace</a>	0
<a href="#">Method make</a>	0
<a href="#">Class JSONWebService</a>	0
<a href="#">Class Constant ControllerNamespace</a>	0
<a href="#">Method make</a>	0
<a href="#">Class ListController</a>	0
<a href="#">Class Constant ControllerNamespace</a>	0
<a href="#">Class Constant DefaultBaseController</a>	0
<a href="#">Class Constant ModelNamespace</a>	0
<a href="#">Method make</a>	0
<a href="#">Class MakeBase</a>	0
<a href="#">Method export</a>	0
<a href="#">Method make</a>	0
<a href="#">Class Model</a>	0
<a href="#">Class Constant ModelNamespace</a>	0
<a href="#">Method make</a>	0
<a href="#">Class PageController</a>	0
<a href="#">Class Constant ControllerNamespace</a>	0
<a href="#">Class Constant DefaultBaseController</a>	0
<a href="#">Method make</a>	0
<a href="#">Class RESTWebService</a>	0
<a href="#">Class Constant ControllerNamespace</a>	0
<a href="#">Method make</a>	0
<a href="#">Class SOAPWebService</a>	0
<a href="#">Class Constant ControllerNamespace</a>	0
<a href="#">Method make</a>	0

<a href="#">migrate.class.php</a>	0
<a href="#">migrationbase.class.php</a>	0
<a href="#">migrationcompare.class.php</a>	0
<a href="#">migrations.class.php</a>	0
<a href="#">test.class.php</a>	0
<a href="#">Class Migrate</a>	0
<a href="#">Method execute</a>	0
<a href="#">Method getOptions</a>	0
<a href="#">Method handleError</a>	0
<a href="#">Method handleException</a>	0
<a href="#">Class MigrationBase</a>	0
<a href="#">Var \$db</a>	0
<a href="#">Var \$version</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method down</a>	0
<a href="#">Method up</a>	0
<a href="#">Class MigrationCompare</a>	0
<a href="#">Method compareVersion</a>	0
<a href="#">Class Migrations</a>	0
<a href="#">Method downgrade</a>	0
<a href="#">Method getCurrentVersion</a>	0
<a href="#">Method to</a>	0
<a href="#">Method upgrade</a>	0
<a href="#">Class Test</a>	0
<a href="#">Method execute</a>	0
<a href="#">Method handleError</a>	0
<a href="#">Method handleException</a>	0
<a href="#">authentication.class.php</a>	0
<a href="#">authenticationstatus.class.php</a>	0
<a href="#">basicauthentication.class.php</a>	0
<a href="#">credentialbase.class.php</a>	0
<a href="#">formsauthentication.class.php</a>	0
<a href="#">roles.class.php</a>	0
<a href="#">tablecredential.class.php</a>	0
<a href="#">usercredential.class.php</a>	0
<a href="#">webserviceauthentication.class.php</a>	0
<a href="#">Class Authentication</a>	0
<a href="#">Var \$identity</a>	0
<a href="#">Var \$isProtected</a>	0
<a href="#">Method authenticate</a>	0
<a href="#">Method authenticated</a>	0
<a href="#">Method authorized</a>	0
<a href="#">Method generateHash</a>	0
<a href="#">Method generateSalt</a>	0
<a href="#">Method getAuthMethod</a>	0
<a href="#">Method isProtected</a>	0
<a href="#">Method redirectToLogin</a>	0
<a href="#">Method signout</a>	0
<a href="#">Class AuthenticationStatus</a>	0

<a href="#">Constructor</a>	<a href="#">construct</a>	0
<a href="#">Method</a>	<a href="#">authenticated</a>	0
<a href="#">Method</a>	<a href="#">disabled</a>	0
<a href="#">Method</a>	<a href="#">invalidCredentials</a>	0
<a href="#">Method</a>	<a href="#">lockedOut</a>	0
<a href="#">Class</a>	<a href="#">BasicAuthentication</a>	0
<a href="#">Method</a>	<a href="#">authenticated</a>	0
<a href="#">Method</a>	<a href="#">getAuthUser</a>	0
<a href="#">Method</a>	<a href="#">isAuthUserSet</a>	0
<a href="#">Method</a>	<a href="#">sendAuthHeaders</a>	0
<a href="#">Method</a>	<a href="#">signout</a>	0
<a href="#">Class</a>	<a href="#">CredentialBase</a>	0
<a href="#">Var</a>	<a href="#">\$credential</a>	0
<a href="#">Constructor</a>	<a href="#">construct</a>	0
<a href="#">Method</a>	<a href="#">authenticate</a>	0
<a href="#">Method</a>	<a href="#">comparePassword</a>	0
<a href="#">Class</a>	<a href="#">FormsAuthentication</a>	0
<a href="#">Method</a>	<a href="#">authenticated</a>	0
<a href="#">Method</a>	<a href="#">authenticateSecret</a>	0
<a href="#">Method</a>	<a href="#">getAuthCookie</a>	0
<a href="#">Method</a>	<a href="#">getAuthSecret</a>	0
<a href="#">Method</a>	<a href="#">isAuthCookieSet</a>	0
<a href="#">Method</a>	<a href="#">redirectFromLoginPage</a>	0
<a href="#">Method</a>	<a href="#">redirectToLoginPage</a>	0
<a href="#">Method</a>	<a href="#">setAuthCookie</a>	0
<a href="#">Method</a>	<a href="#">signout</a>	0
<a href="#">Class</a>	<a href="#">Roles</a>	0
<a href="#">Method</a>	<a href="#">getRolesForUser</a>	0
<a href="#">Method</a>	<a href="#">isUserInRole</a>	0
<a href="#">Method</a>	<a href="#">isUserInRoles</a>	0
<a href="#">Class</a>	<a href="#">TableCredential</a>	0
<a href="#">Method</a>	<a href="#">authenticate</a>	0
<a href="#">Method</a>	<a href="#">comparePassword</a>	0
<a href="#">Method</a>	<a href="#">generateHash</a>	0
<a href="#">Class</a>	<a href="#">UserCredential</a>	0
<a href="#">Method</a>	<a href="#">authenticate</a>	0
<a href="#">Method</a>	<a href="#">comparePassword</a>	0
<a href="#">Class</a>	<a href="#">WebServiceAuthentication</a>	0
<a href="#">Method</a>	<a href="#">authenticated</a>	0
<a href="#">Method</a>	<a href="#">authenticateSecret</a>	0
<a href="#">Method</a>	<a href="#">getAuthSecret</a>	0
<a href="#">Method</a>	<a href="#">getAuthUser</a>	0
<a href="#">Method</a>	<a href="#">isAuthUserSet</a>	0
<a href="#">Method</a>	<a href="#">setAuthUser</a>	0
<a href="#">Method</a>	<a href="#">signout</a>	0
<a href="#">controllertestcasebase.class.php</a>		0
<a href="#">htmltestreporter.class.php</a>		0
<a href="#">migrationtestcasebase.class.php</a>		0
<a href="#">testcasebase.class.php</a>		0

<a href="#">testreporter.class.php</a>	0
<a href="#">unittestcasebase.class.php</a>	0
<a href="#">Class ControllerTestCaseBase</a>	0
<a href="#">Var \$controller</a>	0
<a href="#">Var \$view</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method assertAssigned</a>	0
<a href="#">Method assertMessage</a>	0
<a href="#">Method assertNoResponse</a>	0
<a href="#">Method assertRedirectedTo</a>	0
<a href="#">Method assertResponse</a>	0
<a href="#">Method get</a>	0
<a href="#">Method post</a>	0
<a href="#">Method response</a>	0
<a href="#">Method responseAsXMLEntity</a>	0
<a href="#">Method setUp</a>	0
<a href="#">Method submit</a>	0
<a href="#">Method tearDown</a>	0
<a href="#">Class HTMLTestReporter</a>	0
<a href="#">Method paintException</a>	0
<a href="#">Method paintFooter</a>	0
<a href="#">Method paintHeader</a>	0
<a href="#">Method setElapsedTime</a>	0
<a href="#">Class MigrationsTestCaseBase</a>	0
<a href="#">Var \$state</a>	0
<a href="#">Method setUp</a>	0
<a href="#">Method tearDown</a>	0
<a href="#">Method xxx</a>	0
<a href="#">Class TestCaseBase</a>	0
<a href="#">Var \$fixtures</a>	0
<a href="#">Method cleanup</a>	0
<a href="#">Method getFixtureVariables</a>	0
<a href="#">Method loadCSVFixture</a>	0
<a href="#">Method loadFixtures</a>	0
<a href="#">Method loadSQLFixture</a>	0
<a href="#">Method loadXMLFixture</a>	0
<a href="#">Method prepare</a>	0
<a href="#">Method run</a>	0
<a href="#">Class TestReporter</a>	0
<a href="#">Method setElapsedTime</a>	0
<a href="#">Class UnitTestCaseBase</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method setUp</a>	0
<a href="#">Method tearDown</a>	0
<a href="#">timer.class.php</a>	0
<a href="#">Class Timer</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method elapsed</a>	0
<a href="#">Method pause</a>	0



<a href="#">Method reset</a>	0
<a href="#">Method resume</a>	0
<a href="#">Method start</a>	0
<a href="#">Method stop</a>	0
<a href="#">booleanvalidator.class.php</a>	0
<a href="#">datetimevalidator.class.php</a>	0
<a href="#">emailvalidator.class.php</a>	0
<a href="#">enumvalidator.class.php</a>	0
<a href="#">filesizevalidator.class.php</a>	0
<a href="#">filetypevalidator.class.php</a>	0
<a href="#">integervalidator.class.php</a>	0
<a href="#">lengthvalidator.class.php</a>	0
<a href="#">matchvalidator.class.php</a>	0
<a href="#">numericvalidator.class.php</a>	0
<a href="#">patternvalidator.class.php</a>	0
<a href="#">rangevalidator.class.php</a>	0
<a href="#">requiredvalidator.class.php</a>	0
<a href="#">uniquevalidator.class.php</a>	0
<a href="#">urlvalidator.class.php</a>	0
<a href="#">validatorbase.class.php</a>	0
<a href="#">validatorcollection.class.php</a>	0
<a href="#">Class BooleanValidator</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class DateTimeValidator</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class EmailValidator</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Class EnumValidator</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class FileSizeValidator</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class FileTypeValidator</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method addType</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class IntegerValidator</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class LengthValidator</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method onLoad</a>	0

<a href="#">Method validate</a>	0
<a href="#">Class MatchValidator</a>	0
<a href="#">Var \$controlToMatch</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class NumericValidator</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class PatternValidator</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class RangeValidator</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class RequiredValidator</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class UniqueValidator</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class URLValidator</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Class ValidatorBase</a>	0
<a href="#">Var \$controlToValidate</a>	0
<a href="#">Var \$errorMessage</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method setControlToValidate</a>	0
<a href="#">Method setErrorMessage</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class ValidatorCollection</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method add</a>	0
<a href="#">Method contains</a>	0
<a href="#">controllerbase.class.php</a>	0
<a href="#">formajaxpostevent.class.php</a>	0
<a href="#">formajaxposteventhandler.class.php</a>	0
<a href="#">formpostevent.class.php</a>	0
<a href="#">formposteventhandler.class.php</a>	0
<a href="#">gridviewcolumnajaxpostevent.class.php</a>	0
<a href="#">gridviewcolumnajaxposteventhandler.class.php</a>	0
<a href="#">gridviewcolumnpostevent.class.php</a>	0
<a href="#">gridviewcolumnposteventhandler.class.php</a>	0
<a href="#">inputajaxchangeevent.class.php</a>	0
<a href="#">inputajaxchangeeventhandler.class.php</a>	0

<a href="#">inputajaxpostevent.class.php</a>	0
<a href="#">inputajaxposteventhandler.class.php</a>	0
<a href="#">inputchangeevent.class.php</a>	0
<a href="#">inputchangeeventhandler.class.php</a>	0
<a href="#">inputpostevent.class.php</a>	0
<a href="#">inputposteventhandler.class.php</a>	0
<a href="#">loginformsubmitevent.class.php</a>	0
<a href="#">loginformsubmiteventhandler.class.php</a>	0
<a href="#">masterviewinitevent.class.php</a>	0
<a href="#">masterviewiniteventhandler.class.php</a>	0
<a href="#">masterviewloadevent.class.php</a>	0
<a href="#">masterviewloadeventhandler.class.php</a>	0
<a href="#">pagepostevent.class.php</a>	0
<a href="#">pageposteventhandler.class.php</a>	0
<a href="#">pagerequestevent.class.php</a>	0
<a href="#">pagerequesteventhandler.class.php</a>	0
<a href="#">reportdetailsevent.class.php</a>	0
<a href="#">reportdetailseventhandler.class.php</a>	0
<a href="#">reportfooterevent.class.php</a>	0
<a href="#">reportfootereventhandler.class.php</a>	0
<a href="#">reportgroupfooterevent.class.php</a>	0
<a href="#">reportgroupfootereventhandler.class.php</a>	0
<a href="#">reportgroupheaderevent.class.php</a>	0
<a href="#">reportgroupheadereventhandler.class.php</a>	0
<a href="#">reportheaderevent.class.php</a>	0
<a href="#">reportheadereventhandler.class.php</a>	0
<a href="#">webcontrolcreateevent.class.php</a>	0
<a href="#">webcontrolcreateeventhandler.class.php</a>	0
<a href="#">webcontrolinitevent.class.php</a>	0
<a href="#">webcontroliniteventhandler.class.php</a>	0
<a href="#">webcontrolloadevent.class.php</a>	0
<a href="#">webcontrolloadeventhandler.class.php</a>	0
<a href="#">webcontrolprerenderevent.class.php</a>	0
<a href="#">webcontrolprerendereventhandler.class.php</a>	0
<a href="#">formmodelbase.class.php</a>	0
<a href="#">forwardmethodtype.class.php</a>	0
<a href="#">pagecontrollerbase.class.php</a>	0
<a href="#">jsonservicebase.class.php</a>	0
<a href="#">restwebservicebase.class.php</a>	0
<a href="#">soapwebservicebase.class.php</a>	0
<a href="#">webservicebase.class.php</a>	0
<a href="#">webservicemethod.class.php</a>	0
<a href="#">button.class.php</a>	0
<a href="#">checkbox.class.php</a>	0
<a href="#">checkboxlist.class.php</a>	0
<a href="#">dateselector.class.php</a>	0
<a href="#">datetimeselector.class.php</a>	0
<a href="#">dropdownlist.class.php</a>	0
<a href="#">fieldset.class.php</a>	0

<a href="#">filebrowser.class.php</a>	0
<a href="#">form.class.php</a>	0
<a href="#">gridview.class.php</a>	0
<a href="#">gridviewbutton.class.php</a>	0
<a href="#">gridviewcheckbox.class.php</a>	0
<a href="#">gridviewcolumn.class.php</a>	0
<a href="#">gridviewcolumncollection.class.php</a>	0
<a href="#">gridviewcontrolbase.class.php</a>	0
<a href="#">gridviewdropdownmenu.class.php</a>	0
<a href="#">gridviewimage.class.php</a>	0
<a href="#">gridviewlink.class.php</a>	0
<a href="#">gridviewtextbox.class.php</a>	0
<a href="#">hyperlink.class.php</a>	0
<a href="#">inputbase.class.php</a>	0
<a href="#">label.class.php</a>	0
<a href="#">listbox.class.php</a>	0
<a href="#">listcontrolbase.class.php</a>	0
<a href="#">listitemcollection.class.php</a>	0
<a href="#">loginform.class.php</a>	0
<a href="#">masterview.class.php</a>	0
<a href="#">page.class.php</a>	0
<a href="#">radiobutton.class.php</a>	0
<a href="#">radiobuttonlist.class.php</a>	0
<a href="#">radiogroup.class.php</a>	0
<a href="#">reportgroup.class.php</a>	0
<a href="#">reportview.class.php</a>	0
<a href="#">textarea.class.php</a>	0
<a href="#">textbox.class.php</a>	0
<a href="#">timeselector.class.php</a>	0
<a href="#">treenode.class.php</a>	0
<a href="#">treenodecollection.class.php</a>	0
<a href="#">treeview.class.php</a>	0
<a href="#">view.class.php</a>	0
<a href="#">webcontrolattributecollection.class.php</a>	0
<a href="#">webcontrolbase.class.php</a>	0
<a href="#">webcontrolcollection.class.php</a>	0
<a href="#">Class Button</a>	0
<a href="#">Var \$disableOnSubmit</a>	0
<a href="#">Var \$src</a>	0
<a href="#">Var \$submitText</a>	0
<a href="#">Var \$text</a>	0
<a href="#">Constructor   construct</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Class CheckBox</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method onRequest</a>	0

<a href="#">Class CheckBoxList</a>	0
<a href="#">Var \$multiple</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Class ControllerBase</a>	0
<a href="#">Var \$allowRoles</a>	0
<a href="#">Var \$controllerId</a>	0
<a href="#">Var \$denyRoles</a>	0
<a href="#">Var \$events</a>	0
<a href="#">Var \$outputCache</a>	0
<a href="#">Var \$requireSSL</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method getCached</a>	0
<a href="#">Method getRoles</a>	0
<a href="#">Method getView</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Class DateSelector</a>	0
<a href="#">Var \$allowNull</a>	0
<a href="#">Var \$yearMax</a>	0
<a href="#">Var \$yearMin</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Class DateTimeSelector</a>	0
<a href="#">Var \$allowNull</a>	0
<a href="#">Var \$yearMax</a>	0
<a href="#">Var \$yearMin</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Class DropDownList</a>	0
<a href="#">Var \$listSize</a>	0
<a href="#">Var \$multiple</a>	0
<a href="#">Class Fieldset</a>	0
<a href="#">Var \$legend</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method add</a>	0
<a href="#">Method fillDataSource</a>	0
<a href="#">Method focus</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method readDataSource</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class FileBrowser</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Method getFileInfo</a>	0
<a href="#">Method getFileRawData</a>	0

<a href="#">Method onLoad</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Class Form</a>	0
<a href="#">Var \$action</a>	0
<a href="#">Var \$ajaxPostBack</a>	0
<a href="#">Var \$ajaxValidation</a>	0
<a href="#">Var \$autoFocus</a>	0
<a href="#">Var \$encodeType</a>	0
<a href="#">Var \$forward</a>	0
<a href="#">Var \$hiddenField</a>	0
<a href="#">Var \$method</a>	0
<a href="#">Var \$parameters</a>	0
<a href="#">Var \$submitted</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method add</a>	0
<a href="#">Method addParameter</a>	0
<a href="#">Method begin</a>	0
<a href="#">Method end</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Method getFormDomObject</a>	0
<a href="#">Method onDataBind</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method onPost</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Method save</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class FormAjaxPostEvent</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class FormAjaxPostEventHandler</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class FormModelBase</a>	0
<a href="#">Var \$fields</a>	0
<a href="#">Var \$field_mappings</a>	0
<a href="#">Var \$rules</a>	0
<a href="#">Var \$rule_mappings</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method form</a>	0
<a href="#">Method registerFieldType</a>	0
<a href="#">Method registerRuleType</a>	0
<a href="#">Method save</a>	0
<a href="#">Method toArray</a>	0
<a href="#">Class FormPostEvent</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class FormPostEventHandler</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class ForwardMethodType</a>	0
<a href="#">Method Request</a>	0
<a href="#">Method URI</a>	0
<a href="#">Class GridView</a>	0

<a href="#">Var \$actions</a>	0
<a href="#">Var \$ajaxPostBack</a>	0
<a href="#">Var \$autoGenerateColumns</a>	0
<a href="#">Var \$canChangeOrder</a>	0
<a href="#">Var \$canSort</a>	0
<a href="#">Var \$caption</a>	0
<a href="#">Var \$columns</a>	0
<a href="#">Var \$filterGroupSize</a>	0
<a href="#">Var \$filters</a>	0
<a href="#">Var \$filterValues</a>	0
<a href="#">Var \$listName</a>	0
<a href="#">Var \$moveCursor</a>	0
<a href="#">Var \$moveOrder</a>	0
<a href="#">Var \$multiple</a>	0
<a href="#">Var \$onclick</a>	0
<a href="#">Var \$ondblclick</a>	0
<a href="#">Var \$onmouseout</a>	0
<a href="#">Var \$onmouseover</a>	0
<a href="#">Var \$orderByField</a>	0
<a href="#">Var \$page</a>	0
<a href="#">Var \$pageSize</a>	0
<a href="#">Var \$selected</a>	0
<a href="#">Var \$showFilters</a>	0
<a href="#">Var \$showFooter</a>	0
<a href="#">Var \$showHeader</a>	0
<a href="#">Var \$showList</a>	0
<a href="#">Var \$showPageNumber</a>	0
<a href="#">Var \$showPrimaryKey</a>	0
<a href="#">Var \$sortBy</a>	0
<a href="#">Var \$sortOrder</a>	0
<a href="#">Var \$valueField</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method addColumn</a>	0
<a href="#">Method findColumn</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Method getPagination</a>	0
<a href="#">Method getRowBody</a>	0
<a href="#">Method getRowFilter</a>	0
<a href="#">Method getRowFooter</a>	0
<a href="#">Method getRowHeader</a>	0
<a href="#">Method onDataBind</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method onLoadViewState</a>	0
<a href="#">Method onPost</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Method onSaveViewState</a>	0
<a href="#">Method onUpdateAjax</a>	0
<a href="#">Method setFilterValues</a>	0
<a href="#">Class GridViewButton</a>	0

<a href="#">Var \$scanFilter</a>	0
<a href="#">Var \$confirmation</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method getItemText</a>	0
<a href="#">Method onPost</a>	0
<a href="#">Class GridViewCheckBox</a>	0
<a href="#">Method getItemText</a>	0
<a href="#">Class GridViewColumn</a>	0
<a href="#">Var \$scanFilter</a>	0
<a href="#">Var \$className</a>	0
<a href="#">Var \$dataField</a>	0
<a href="#">Var \$events</a>	0
<a href="#">Var \$footerText</a>	0
<a href="#">Var \$headerText</a>	0
<a href="#">Var \$itemText</a>	0
<a href="#">Var \$onclick</a>	0
<a href="#">Var \$ondblclick</a>	0
<a href="#">Var \$onmouseout</a>	0
<a href="#">Var \$onmouseover</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method getRequestData</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method onPost</a>	0
<a href="#">Method onRender</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Class GridViewColumnAjaxPostEvent</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class GridViewColumnAjaxPostEventHandler</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class GridViewColumnCollection</a>	0
<a href="#">Method add</a>	0
<a href="#">Method addBefore</a>	0
<a href="#">Method findColumn</a>	0
<a href="#">Method itemAt</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method onPost</a>	0
<a href="#">Method onRender</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Class GridViewColumnPostEvent</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class GridViewColumnPostEventHandler</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Class GridViewControlBase</a>	0
<a href="#">Var \$ajaxPostBack</a>	0
<a href="#">Var \$escapeOutput</a>	0
<a href="#">Var \$parameter</a>	0
<a href="#">Var \$pkey</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method escape</a>	0



<a href="#">Method getItemText</a>	0
<a href="#">Method onPost</a>	0
<a href="#">Method onRender</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Class GridViewDropDownMenu</a>	0
<a href="#">Var \$items</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method getItemText</a>	0
<a href="#">Class GridViewImage</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class GridViewLink</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class GridViewTextBox</a>	0
<a href="#">Var \$size</a>	0
<a href="#">Method getItemText</a>	0
<a href="#">Class HyperLink</a>	0
<a href="#">Var \$src</a>	0
<a href="#">Var \$target</a>	0
<a href="#">Var \$text</a>	0
<a href="#">Var \$url</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Class InputAjaxChangeEvent</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class InputAjaxChangeEventHandler</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class InputAjaxPostEvent</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class InputAjaxPostEventHandler</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class InputBase</a>	0
<a href="#">Var \$ajaxPostBack</a>	0
<a href="#">Var \$ajaxValidation</a>	0
<a href="#">Var \$autoPostBack</a>	0
<a href="#">Var \$changed</a>	0
<a href="#">Var \$dataField</a>	0
<a href="#">Var \$defaultHTMLControlId</a>	0
<a href="#">Var \$disabled</a>	0
<a href="#">Var \$enableViewState</a>	0
<a href="#">Var \$label</a>	0
<a href="#">Var \$readonly</a>	0
<a href="#">Var \$submitted</a>	0
<a href="#">Var \$tabIndex</a>	0
<a href="#">Var \$tooltip</a>	0
<a href="#">Var \$validators</a>	0
<a href="#">Var \$value</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method addValidator</a>	0
<a href="#">Method error</a>	0

<a href="#">Method fetchError</a>	0
<a href="#">Method fillDataSource</a>	0
<a href="#">Method focus</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Method getInputDomObject</a>	0
<a href="#">Method onDataBind</a>	0
<a href="#">Method onInit</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method onLoadViewState</a>	0
<a href="#">Method onPost</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Method onSaveViewState</a>	0
<a href="#">Method onUpdateAjax</a>	0
<a href="#">Method readDataSource</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class InputChangeEvent</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class InputChangeEventHandler</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class InputPostEvent</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class InputPostEventHandler</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class JSONServiceBase</a>	0
<a href="#">Var \$options</a>	0
<a href="#">Method configure</a>	0
<a href="#">Method handle</a>	0
<a href="#">Class Label</a>	0
<a href="#">Var \$text</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Class ListBox</a>	0
<a href="#">Var \$listSize</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Method onUpdateAjax</a>	0
<a href="#">Class ListControlBase</a>	0
<a href="#">Var \$items</a>	0
<a href="#">Var \$multiple</a>	0
<a href="#">Var \$textField</a>	0
<a href="#">Var \$valueField</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method onDataBind</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method onLoadViewState</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class ListItemCollection</a>	0
<a href="#">Method add</a>	0
<a href="#">Method containsItem</a>	0

<a href="#">Method indexOf</a>	0
<a href="#">Method indexOfItem</a>	0
<a href="#">Class LoginForm</a>	0
<a href="#">Var \$disabledMsg</a>	0
<a href="#">Var \$emailFromAddress</a>	0
<a href="#">Var \$emailNotFoundMsg</a>	0
<a href="#">Var \$emailSubject</a>	0
<a href="#">Var \$invalidCredentialsMsg</a>	0
<a href="#">Var \$lockoutMsg</a>	0
<a href="#">Var \$mailClient</a>	0
<a href="#">Var \$passwordResetMsg</a>	0
<a href="#">Var \$passwordResetSentMsg</a>	0
<a href="#">Var \$successMsg</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Method onInit</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Class LoginFormSubmitEvent</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class LoginFormSubmitEventHandler</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class MasterView</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method content</a>	0
<a href="#">Method getContentAreaIdString</a>	0
<a href="#">Method handleMasterViewPostEvents</a>	0
<a href="#">Method initMasterView</a>	0
<a href="#">Method loadMasterView</a>	0
<a href="#">Method loadMasterViewViewState</a>	0
<a href="#">Method masterViewRequestProcessor</a>	0
<a href="#">Method saveMasterViewViewState</a>	0
<a href="#">Class MasterViewInitEvent</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class MasterViewInitEventHandler</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class MasterViewLoadEvent</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class MasterViewLoadEventHandler</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class Page</a>	0
<a href="#">Var \$contentType</a>	0
<a href="#">Var \$linkElements</a>	0
<a href="#">Var \$metaElements</a>	0
<a href="#">Var \$objectElements</a>	0
<a href="#">Var \$onload</a>	0
<a href="#">Var \$scriptElements</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method addLink</a>	0
<a href="#">Method addMetaTag</a>	0

<a href="#">Method addScript</a>	0
<a href="#">Method clearAjaxBuffer</a>	0
<a href="#">Method fetch</a>	0
<a href="#">Method fetchPageHeader</a>	0
<a href="#">Method getAjaxBuffer</a>	0
<a href="#">Method isAjaxBufferReady</a>	0
<a href="#">Method loadAjaxBuffer</a>	0
<a href="#">Method loadAjaxJScriptBuffer</a>	0
<a href="#">Method onInit</a>	0
<a href="#">Method onPost</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Class PageControllerBase</a>	0
<a href="#">Var \$page</a>	0
<a href="#">Var \$theme</a>	0
<a href="#">Method getView</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Class PagePostEvent</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class PagePostEventHandler</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class PageRequestEvent</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class PageRequestEventHandler</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class RadioButton</a>	0
<a href="#">Var \$checked</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Class RadioButtonList</a>	0
<a href="#">Var \$multiple</a>	0
<a href="#">Var \$onclick</a>	0
<a href="#">Var \$ondblclick</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Class RadioGroup</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method add</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Class ReportDetailsEvent</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class ReportDetailsEventHandler</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class ReportFooterEvent</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class ReportFooterEventHandler</a>	0

<a href="#">Constructor</a>	<a href="#">construct</a>	0
<a href="#">Class ReportGroup</a>		0
<a href="#">Constructor</a>	<a href="#">construct</a>	0
<a href="#">Class ReportGroupFooterEvent</a>		0
<a href="#">Constructor</a>	<a href="#">construct</a>	0
<a href="#">Class ReportGroupFooterEventHandler</a>		0
<a href="#">Constructor</a>	<a href="#">construct</a>	0
<a href="#">Class ReportGroupHeaderEvent</a>		0
<a href="#">Constructor</a>	<a href="#">construct</a>	0
<a href="#">Class ReportGroupHeaderEventHandler</a>		0
<a href="#">Constructor</a>	<a href="#">construct</a>	0
<a href="#">Class ReportHeaderEvent</a>		0
<a href="#">Constructor</a>	<a href="#">construct</a>	0
<a href="#">Class ReportHeaderEventHandler</a>		0
<a href="#">Constructor</a>	<a href="#">construct</a>	0
<a href="#">Class ReportView</a>		0
<a href="#">Constructor</a>	<a href="#">construct</a>	0
<a href="#">Method</a>	<a href="#">addFilter</a>	0
<a href="#">Method</a>	<a href="#">addGrouping</a>	0
<a href="#">Method</a>	<a href="#">addSorting</a>	0
<a href="#">Method</a>	<a href="#">fetch</a>	0
<a href="#">Method</a>	<a href="#">getDomObject</a>	0
<a href="#">Method</a>	<a href="#">getGrouping</a>	0
<a href="#">Method</a>	<a href="#">onDataBind</a>	0
<a href="#">Class RESTWebServiceBase</a>		0
<a href="#">Var</a>	<a href="#">\$options</a>	0
<a href="#">Method</a>	<a href="#">configure</a>	0
<a href="#">Method</a>	<a href="#">delete</a>	0
<a href="#">Method</a>	<a href="#">get</a>	0
<a href="#">Method</a>	<a href="#">handle</a>	0
<a href="#">Method</a>	<a href="#">post</a>	0
<a href="#">Method</a>	<a href="#">put</a>	0
<a href="#">Class SOAPWebServiceBase</a>		0
<a href="#">Var</a>	<a href="#">\$cache</a>	0
<a href="#">Var</a>	<a href="#">\$encoding</a>	0
<a href="#">Var</a>	<a href="#">\$namespace</a>	0
<a href="#">Var</a>	<a href="#">\$version</a>	0
<a href="#">Var</a>	<a href="#">\$xreserved_methods</a>	0
<a href="#">Method</a>	<a href="#">configure</a>	0
<a href="#">Method</a>	<a href="#">handle</a>	0
<a href="#">Class TextArea</a>		0
<a href="#">Var</a>	<a href="#">\$multiline</a>	0
<a href="#">Class TextBox</a>		0
<a href="#">Var</a>	<a href="#">\$cols</a>	0
<a href="#">Var</a>	<a href="#">\$disableAutoComplete</a>	0
<a href="#">Var</a>	<a href="#">\$disableEnterKey</a>	0
<a href="#">Var</a>	<a href="#">\$mask</a>	0
<a href="#">Var</a>	<a href="#">\$maxLength</a>	0
<a href="#">Var</a>	<a href="#">\$multiline</a>	0

<a href="#">Var \$rows</a>	0
<a href="#">Var \$size</a>	0
<a href="#">Var \$watermark</a>	0
<a href="#">Method fillDataSet</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Class TimeSelector</a>	0
<a href="#">Var \$allowNull</a>	0
<a href="#">Method getDomObject</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Class TreeNode</a>	0
<a href="#">Var \$childNodes</a>	0
<a href="#">Var \$expanded</a>	0
<a href="#">Var \$id</a>	0
<a href="#">Var \$imgSrc</a>	0
<a href="#">Var \$parentNode</a>	0
<a href="#">Var \$textOrHtml</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method addChild</a>	0
<a href="#">Method collapse</a>	0
<a href="#">Method collapseAll</a>	0
<a href="#">Method createChild</a>	0
<a href="#">Method expand</a>	0
<a href="#">Method expandAll</a>	0
<a href="#">Method findChildNode</a>	0
<a href="#">Method getAncestry</a>	0
<a href="#">Method getChild</a>	0
<a href="#">Method getChildren</a>	0
<a href="#">Method getParent</a>	0
<a href="#">Method getSiblings</a>	0
<a href="#">Method hasChild</a>	0
<a href="#">Method hasChildNode</a>	0
<a href="#">Method loadViewState</a>	0
<a href="#">Method requestProcessor</a>	0
<a href="#">Method saveViewState</a>	0
<a href="#">Method setParent</a>	0
<a href="#">Class TreeNodeCollection</a>	0
<a href="#">Method add</a>	0
<a href="#">Method contains</a>	0
<a href="#">Method indexOf</a>	0
<a href="#">Method itemAt</a>	0
<a href="#">Class TreeView</a>	0
<a href="#">Var \$rootNode</a>	0
<a href="#">Var \$showIcons</a>	0
<a href="#">Var \$showRoot</a>	0
<a href="#">Method findTreeItem</a>	0
<a href="#">Method getDomObject</a>	0

<a href="#">Method <code>getNodeDomObject</code></a>	0
<a href="#">Method <code>onLoad</code></a>	0
<a href="#">Method <code>onLoadViewState</code></a>	0
<a href="#">Method <code>onRequest</code></a>	0
<a href="#">Method <code>onSaveViewState</code></a>	0
<a href="#">Class <code>View</code></a>	0
<a href="#">Var <code>\$charset</code></a>	0
<a href="#">Var <code>\$contentType</code></a>	0
<a href="#">Var <code>\$template</code></a>	0
<a href="#">Constructor <code>construct</code></a>	0
<a href="#">Method <code>add</code></a>	0
<a href="#">Method <code>addHeader</code></a>	0
<a href="#">Method <code>assign</code></a>	0
<a href="#">Method <code>fetch</code></a>	0
<a href="#">Method <code>fetchTemplate</code></a>	0
<a href="#">Method <code>getNodeDomObject</code></a>	0
<a href="#">Method <code>onInit</code></a>	0
<a href="#">Method <code>onLoad</code></a>	0
<a href="#">Method <code>onPost</code></a>	0
<a href="#">Method <code>onRequest</code></a>	0
<a href="#">Method <code>render</code></a>	0
<a href="#">Method <code>sendHeaders</code></a>	0
<a href="#">Method <code>setData</code></a>	0
<a href="#">Method <code>setMaster</code></a>	0
<a href="#">Class <code>WebControlAttributeCollection</code></a>	0
<a href="#">Method <code>append</code></a>	0
<a href="#">Class <code>WebControlBase</code></a>	0
<a href="#">Var <code>\$ajaxCallback</code></a>	0
<a href="#">Var <code>\$ajaxEventHandler</code></a>	0
<a href="#">Var <code>\$ajaxHttpRequest</code></a>	0
<a href="#">Var <code>\$attributes</code></a>	0
<a href="#">Var <code>\$controls</code></a>	0
<a href="#">Var <code>\$dataSource</code></a>	0
<a href="#">Var <code>\$enableViewState</code></a>	0
<a href="#">Var <code>\$escapeOutput</code></a>	0
<a href="#">Var <code>\$events</code></a>	0
<a href="#">Var <code>\$visible</code></a>	0
<a href="#">Constructor <code>construct</code></a>	0
<a href="#">Method <code>addControl</code></a>	0
<a href="#">Method <code>bind</code></a>	0
<a href="#">Method <code>createDomObject</code></a>	0
<a href="#">Method <code>escape</code></a>	0
<a href="#">Method <code>fetch</code></a>	0
<a href="#">Method <code>findControl</code></a>	0
<a href="#">Method <code>getControl</code></a>	0
<a href="#">Method <code>getNodeDomObject</code></a>	0
<a href="#">Method <code>getHTMLControlId</code></a>	0
<a href="#">Method <code>getHTMLControlIdString</code></a>	0
<a href="#">Method <code>getParentByType</code></a>	0

<a href="#">Method getParentString</a>	0
<a href="#">Method getQueryString</a>	0
<a href="#">Method getRequestData</a>	0
<a href="#">Method handlePostEvents</a>	0
<a href="#">Method hasControl</a>	0
<a href="#">Method init</a>	0
<a href="#">Method isControl</a>	0
<a href="#">Method load</a>	0
<a href="#">Method loadViewState</a>	0
<a href="#">Method onBind</a>	0
<a href="#">Method onInit</a>	0
<a href="#">Method onLoad</a>	0
<a href="#">Method onLoadViewState</a>	0
<a href="#">Method onPost</a>	0
<a href="#">Method onPreRender</a>	0
<a href="#">Method onRequest</a>	0
<a href="#">Method onSaveViewState</a>	0
<a href="#">Method onUpdateAjax</a>	0
<a href="#">Method preRender</a>	0
<a href="#">Method refreshDataSource</a>	0
<a href="#">Method removeControl</a>	0
<a href="#">Method render</a>	0
<a href="#">Method replaceControl</a>	0
<a href="#">Method requestProcessor</a>	0
<a href="#">Method saveViewState</a>	0
<a href="#">Method setParent</a>	0
<a href="#">Method updateAjax</a>	0
<a href="#">Class WebControlCollection</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method add</a>	0
<a href="#">Method contains</a>	0
<a href="#">Method indexOf</a>	0
<a href="#">Method itemAt</a>	0
<a href="#">Method remove</a>	0
<a href="#">Method replace</a>	0
<a href="#">Class WebControlCreateEvent</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class WebControlCreateEventHandler</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class WebControlInitEvent</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class WebControlInitEventHandler</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class WebControlLoadEvent</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class WebControlLoadEventHandler</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class WebControlPreRenderEvent</a>	0
<a href="#">Constructor construct</a>	0



<a href="#">Class WebControlPreRenderEventHandler</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Class WebServiceBase</a>	0
<a href="#">Var \$remoteProcedures</a>	0
<a href="#">Var \$reserved methods</a>	0
<a href="#">Var \$view</a>	0
<a href="#">Method configure</a>	0
<a href="#">Method getView</a>	0
<a href="#">Method handle</a>	0
<a href="#">Class WebServiceMethod</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method getFunction</a>	0
<a href="#">Method getParameters</a>	0
<a href="#">Method setFunction</a>	0
<a href="#">Method setParameters</a>	0
<a href="#">domobject.class.php</a>	0
<a href="#">textnode.class.php</a>	0
<a href="#">xmlentity.class.php</a>	0
<a href="#">xmlentitycollection.class.php</a>	0
<a href="#">xmlexception.class.php</a>	0
<a href="#">xmlparser.class.php</a>	0
<a href="#">Class DomObject</a>	0
<a href="#">Var \$innerHtml</a>	0
<a href="#">Method getChildByld</a>	0
<a href="#">Method getEntityValue</a>	0
<a href="#">Method getXMLString</a>	0
<a href="#">Method render</a>	0
<a href="#">Class TextNode</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method addChild</a>	0
<a href="#">Method appendAttribute</a>	0
<a href="#">Method getAttribute</a>	0
<a href="#">Method getChild</a>	0
<a href="#">Method getChildByAttribute</a>	0
<a href="#">Method getChildByName</a>	0
<a href="#">Method getChildrenByName</a>	0
<a href="#">Method getXMLString</a>	0
<a href="#">Method removeAttribute</a>	0
<a href="#">Method removeChild</a>	0
<a href="#">Method replaceChild</a>	0
<a href="#">Method setAttribute</a>	0
<a href="#">Class XMLEntity</a>	0
<a href="#">Var \$attributes</a>	0
<a href="#">Var \$cdata</a>	0
<a href="#">Var \$children</a>	0
<a href="#">Var \$forceCloseTag</a>	0
<a href="#">Var \$name</a>	0
<a href="#">Var \$value</a>	0
<a href="#">Constructor construct</a>	0

<a href="#">Method addChild</a>	0
<a href="#">Method appendAttribute</a>	0
<a href="#">Method fetch</a>	0
<a href="#">Method findChildByName</a>	0
<a href="#">Method getAttribute</a>	0
<a href="#">Method getChild</a>	0
<a href="#">Method getChildByAttribute</a>	0
<a href="#">Method getChildByName</a>	0
<a href="#">Method getChildrenByName</a>	0
<a href="#">Method getEntityValue</a>	0
<a href="#">Method getXMLString</a>	0
<a href="#">Method removeAttribute</a>	0
<a href="#">Method removeChild</a>	0
<a href="#">Method replaceChild</a>	0
<a href="#">Method setAttribute</a>	0
<a href="#">Class XMLEntityCollection</a>	0
<a href="#">Method add</a>	0
<a href="#">Method contains</a>	0
<a href="#">Method indexOf</a>	0
<a href="#">Method itemAt</a>	0
<a href="#">Class XMLException</a>	0
<a href="#">Class XMLParser</a>	0
<a href="#">Var \$data</a>	0
<a href="#">Var \$error</a>	0
<a href="#">Method parse</a>	0
<a href="#">Appendix A - Class Trees</a>	0
<a href="#">PHPRum</a>	0

# index.php

**Startup script** This script creates an instance of the application and executes

This should be the only PHP script directly accessible via the web Do not modify this script!

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

include ['../system/base/run.php'](#)*line 19]*

**load framework and application**

# appconfiguration.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# applicationbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# appmessage.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# appmessagetype.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# appstate.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# argumentoutofrangeexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# badmembercallexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# badmethodcallexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# classloader.inc.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

*string* function class\_search(\$className, [\$base = '/']) [*line* [78](#)]

***Function Parameters:***

- *string* **\$className** name of class
- **\$base**

**class\_search**

Returns path to class folder if found

## debug tools

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

*void* function alloc() [[line 112](#)]

### **Print memory usage**

*void* function breakpoint() [[line 26](#)]

### **Set breakpoint**

*string* function callstack() [[line 36](#)]

### **Return callstack string as html**

*void* function clr() [[line 17](#)]

### **Clear all output buffers**

*void* function dmp(\$var, [\$halt = true]) [[line 161](#)]

#### **Function Parameters:**

- *mixed* **\$var** Variable to print
- *bool* **\$halt** Specifies whether to halt application

### **Print variable to screen**

*void* function stp() [[line 144](#)]

### **Stop timer and print results**

*void* function strt() [[line 133](#)]

## **Begin timer**

*void* function trace(\$string, \$var) [[line 175](#)]

### ***Function Parameters:***

- *mixed* **\$var** Variable to trace
- **\$string**

## **Trace**

*void* function trap() [[line 103](#)]

## **begin trapping memory**

# eventbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# eventhandlerbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# exceptionbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# indexoutofrangeexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# invalidargumentexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# invalidoperationexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# methodnotimplementedexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# missingargumentexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# modelbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# rum.php

## Startup script

This script will detect the following environmental variables

__ROOT__	absolute path to the root folder	__HTDOCS_PATH__	absolute path to the web folder
__APP_URI__	root relative URI to the web folder		
__PROTOCOL__	protocol prefix (http https)://	__HOST__	host name
__PORT__	server port		

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

\_\_ACTIVERECORD\_AUTO\_MAP\_\_ = FALSE *[line 263]*  
**specifies the number of warnings to dump**

\_\_APP\_CONF\_FILENAME\_\_ = '/application.xml' *[line 153]*  
**specifies the environment parameter**

\_\_APP\_PATH\_\_ = \_\_ROOT\_\_ . '/app' *[line 283]*  
**specifies the root relative path to the application folder**

\_\_APP\_URI\_\_ = str\_replace(" ", "%20", substr(\$\_SERVER['SCRIPT\_NAME'], 0, strpos(\$\_SERVER['SCRIPT\_NAME'], '/'))) *[line 118]*  
**specifies the application uri**

\_\_ASSETS\_URI\_\_ = \_\_APP\_URI\_\_ . '/assets' *[line 123]*  
**specifies the URI to the assets folder**

\_\_ASYNC\_REQUEST\_PARAMETER\_\_ = 'async' *[line 183]*  
**specifies the async request parameter**



`__BUILD_PATH__ = __ROOT__.'/.build' [line 273]`  
**specifies the root relative path to the cache folder**

`__CACHE_PATH__ = __ROOT__.'/.cache' [line 278]`  
**specifies the root relative path to the cache folder**

`__CACHE_TABLENAME__ = 'cache' [line 138]`  
**specifies the cache table name**

`__CLASS_EXTENSION__ = '.class.php' [line 208]`  
**specifies the class extension**

`__CONFIG_PATH__ = __ROOT__.'/.app/config' [line 288]`  
**specifies the root relative path to the configuration folder**

`__CONTROLLERS_NAMESPACE__ = 'Controllers' [line 168]`  
**specifies the models namespace**

`__CONTROLLERS_PATH__ = __ROOT__.'/.app/controllers' [line 313]`  
**specifies the default controller path**

`__CONTROLLER_EXTENSION__ = '.php' [line 213]`  
**specifies the controller extension**

`__CONTROLLER_TESTCASE_SUFFIX__ = 'ControllerTestCase' [line 233]`  
**specifies the controller testcase suffix**

`__DB_SCHEMA_VERSION_TABLENAME__ = 'db_schema_version' [line 133]`  
**specifies the db\_schema\_version table name**

`__DEFAULT_TIMEZONE__ = 'Canada/Mountain' [line 27]`  
**specifies the default timezone**

`__DEPLOYMENT_EXTENSION__ = '.php' [line 223]`  
**specifies the controller extension**

`__DEPLOY_PATH__ = __CONFIG_PATH__.'/.deploy' [line 298]`  
**specifies the root relative path to the deployment folder**

`__DEV_ENV__ = 'dev' [line 188]`

**specifies the dev environment name**

`__ENV_CONF_FILENAME__ = '/environment.php' [line 158]`

**specifies the environment parameter**

`__ENV_PARAMETER__ = 'APP_ENV' [line 163]`

**specifies the environment parameter**

`__ENV_PATH__ = __CONFIG_PATH__.'/environments' [line 293]`

**specifies the root relative path to the configuration folder**

`__ERROR_LIMIT__ = 10 [line 243]`

**specifies the number of warnings to dump**

`__FIXTURES_PATH__ = __ROOT__.'/app/tests/fixtures' [line 333]`

**specifies the default fixtures path**

`__FUNCTIONAL_TESTS_PATH__ = __ROOT__.'/app/tests/functional' [line 338]`

**specifies the default functional test path**

`__HOST__ = isset($_SERVER['HTTP_HOST'])?$_SERVER['HTTP_HOST']:" [line 113]`

**specifies the host name**

`__HTDOCS_PATH__ =  
substr($_SERVER['SCRIPT_FILENAME'],0,strrpos(str_replace('\\','/',$_SERVER['SCRIPT_FILENAME']),'/')) [line 51]`

**specifies the public web folder path**

`__LANGS_FILE__ = __CONFIG_PATH__.'/langs.xml' [line 303]`

**specifies the path to the langs file**

`__LANGS_TABLENAME__ = 'langs' [line 148]`

**specifies the langs table name**

`__LIB_PATH__ = __SYSTEM_PATH__.'/libs' [line 363]`

**specifies the root relative path to the libs folder**

`__LOGS_TABLENAME__ = 'logs' [line 143]`

**specifies the logs table name**

`__LOG_PATH__ = __ROOT__.'/logs' [line 353]`

**specifies the root relative path to the log folder**

`__MIGRATIONS_PATH__ = __CONFIG_PATH__.'/migrations' [line 308]`

**specifies the root relative path to the migrations folder**

`__MODELS_NAMESPACE__ = 'Models' [line 173]`

**specifies the models namespace**

`__MODELS_PATH__ = __ROOT__.'/app/models' [line 318]`

**specifies the default model path**

`__NAME__ = isset($_SERVER['SERVER_NAME'])?$_SERVER['SERVER_NAME']:" [line 103]`

**specifies the server name**

`__PAGE_EXTENSION__ = '/' [line 203]`

**specifies the request page extension**

`__PAGE_REQUEST_PARAMETER__ = 'page' [line 178]`

**specifies the page request parameter**

`__PLUGINS_PATH__ = __ROOT__.'/app/plugins' [line 323]`

**specifies the root relative path to the plugins folder**

`__PORT__ = isset($_SERVER['SERVER_PORT'])?$_SERVER['SERVER_PORT']:" [line 108]`

**specifies the server port**

`__PROTOCOL__ = $protocol [line 98]`

**specifies the transfer protocol**

`__QUOTE_STYLE__ = ENT_COMPAT [line 248]`

**specifies the quote style when escaping**

`__ROOT__ = $current_dir [line 64]`

`__SSL_PORT__ = 443 [line 198]`

**specifies the SSL port**

`__SYSTEM_PATH__ = __ROOT__.'./system' [line 358]`

**specifies the root relative path to the system folder**

`__TEMPLATES_PATH__ = __ROOT__.'./app/layouts' [line 328]`

**specifies the default template path**

`__TEMPLATE_EXTENSION__ = '.tpl' [line 218]`

**specifies the template extension**

`__TESTCASE_SUFFIX__ = 'TestCase' [line 228]`

**specifies the testcase suffix**

`__TEST_ENV__ = 'test' [line 193]`

**specifies the test environment name**

`__UNIT_TESTS_PATH__ = __ROOT__.'./app/tests/unit' [line 343]`

**specifies the default unit test path**

`__VALIDATION_TIMEOUT__ = '3000' [line 238]`

**specifies the asynchronous validation timeout**

`__VIEWS_PATH__ = __ROOT__.'./app/views' [line 348]`

**specifies the default view path**

`require __SYSTEM_PATH__.'./base/build.class.php' [line 398]`

`require __SYSTEM_PATH__.'./rum.class.php' [line 399]`

`require __SYSTEM_PATH__.'./base/framework.info.file' [line 396]`

`require __SYSTEM_PATH__.'./base/classloader.inc.php' [line 397]`

# typemismatchexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# consoleapplicationbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# loggerbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# make.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# rum.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# tester.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# gridviewchangeorderevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# gridviewchangeordereventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# gridviewfilterevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# gridviewfiltereventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# gridviewsortevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# gridviewsorteventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# webapplicationbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class AppConfiguration

[line 73]

**Provides access to the Application Configuration**

- **Package** PHPRum
- **Property** [AppState](#) \$state: specifies app state
- **Property** string \$lang: specifies default language
- **Property** string \$charset: specifies default character set
- **Property** [AppSettingsCollection](#) \$appsettings: collection of app settings
- **Property** string \$defaultTheme: specifies the default theme
- **Property** bool \$viewStateEnabled: specifies whether viewstate is enabled
- **Property** string \$viewStateMethod: specifies the viewstate method
- **Property** int \$viewStateExpires: specifies the viewstate expiration in seconds form the current time
- **Property** string \$themes: path to themes folder
- **Property** string \$defaultController: specifies default controller
- **Property** string \$requestParameter: specifies page request parameter
- **Property** bool \$rewriteURIS: specifies whether to rewrite URI's

- **Property** bool \$cookielessSession: specifies whether cookieless session is enabled
- **Property** int \$sessionTimeout: specifies the session timeout
- **Property** bool \$cacheEnabled: specifies if cache is enabled
- **Property** int \$cacheExpires: specifies cache expiration in seconds
- **Property** string \$authenticationMethod: specifies authentication method
- **Property** string \$authenticationDeny: specifies pages to deny
- **Property** string \$authenticationAllow: specifies pages to allow
- **Property** string \$authenticationRestrict: specifies allowed ip address
- **Property** int \$authenticationMaxInvalidAttempts: specifies max allowed invalid attempts
- **Property** int \$authenticationAttemptWindow: specifies attempt window in seconds
- **Property** bool \$authenticationRequireSSL: specifies whether to require SSL for authentication
- **Property** string \$authenticationBasicRealm: specifies realm for basic authentication
- **Property** string \$authenticationFormsLoginPage: specifies authentication login page
- **Property** string \$authenticationFormsCookieName: specifies session cookie name
- **Property** string \$authenticationFormsSecret: specifies secret used when checking auth cookies
- **Property** array \$authenticationCredentialsUsers: array of authentication users
- **Property** array \$authenticationCredentialsTables: array of authentication tables
- **Property** array \$authenticationMemberships: array of memberships
- **Property** array \$authenticationMembershipsTables: array of membership tables
- **Property** array \$authorizationDeny: specifies the roles that are denied access by default
- **Property** array \$authorizationAllow: specifies the roles that are granted access by default
- **Property** bool \$authorizationRequireSSL: specifies whether to require SSL for authentication
- **Property** array \$authorizationPages: contains the page authorization configuration
- **Property** string \$dsn: dsn connecting string
- **Property** string \$test\_dsn: testcase dsn connectin string
- **Property** array \$errors: array of errors
- **Property** string \$root: path to root folder
- **Property** string \$htdocs: path to htdocs folder
- **Property** string \$protocol: server protocol

- **Property** string \$host: server host
- **Property** string \$uri: base application URI
- **Property** string \$url: base application URL
- **Property** string \$assets: base URI to the assets folder
- **Property** string \$controllers: path to controllers folder
- **Property** string \$views: path to views folder
- **Property** string \$templates: path to templates folder
- **Property** string \$functionaltests: path to functional test folder
- **Property** string \$unittests: path to unit test folder
- **Property** string \$fixtures: path to fixtures folder
- **Property** string \$system: path to system folder
- **Property** string \$cache: path to cache folder
- **Property** string \$tmp: path to tmp folder
- **Property** string \$logs: path to logs folder
- **Author** Darnell Shinbine

Constructor *void* function AppConfiguration::\_\_construct() [[line 486](#)]

**Creates an instance of the controller and sets default action map.**

- **Access** public

*void* function AppConfiguration::loadAppConfig(\$file) [[line 500](#)]

**Function Parameters:**

- *string* **\$file** path to config file

**loads config data from an xml file**

- **Access** public

## Class ApplicationBase

[line 32]

**This class represents the base application. This class receives command line input and delegates responsibilities to sub components.**

- **Package** PHPRum
- **Property** string \$applicationId: Specifies the application Id
- **Property** string \$namespace: Specifies the application namespace
- **Property** [AppConfiguration](#) \$config: Reference to the AppConfiguration
- **Property** [CacheBase](#) \$cache: Reference to the Cache object
- **Property** [LoggerBase](#) \$logger: Reference to the Logger object
- **Property** [TranslatorBase](#) \$translator: Reference to the Translator object
- **Property** [IMailClient](#) \$mailClient: Reference to the Mail client object
- **Property** string \$lang: Specifies the language
- **Property** string \$charset: Specifies the character set
- **Property** [DataAdapter](#) \$dataAdapter: Reference to the DataAdapter
- **Property** bool \$debug: Specifies whether the application is in debug mode
- **Property** [Timer](#) \$timer: Reference to the app Timer
- **Property** [EventCollection](#) \$events: event collection
- **Author** Darnell Shinbine
- **Abstract Element**

## ApplicationBase::\$events

*EventCollection* = null [[line 110](#)]

### event collection

- **Access** protected

Constructor *void* function ApplicationBase::\_\_construct() [[line 126](#)]

### Constructor

Creates an instance of the controller and sets default action map.

- **Access** public

*void* function ApplicationBase::execute() [[line 419](#)]

### execute the application

- **Abstract Element**
- **Access** protected

*CacheBase* function ApplicationBase::getCache() [[line 450](#)]

**returns the common cache object (overridden this method in the application class)**

- **Access** protected

*ApplicationBase* function ApplicationBase::getInstance([\$initialize = null]) [[line 358](#)]

### **Function Parameters:**

- [ApplicationBase](#) **\$initialize** new instance of an ApplicationBase object

**Returns a static instance of the app server. This provides a way of accessing app variables from outside the class.**

must be initialized by `$controller = ApplicationBase::getInstance( $servletObject );`

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

*LoggerBase* function `ApplicationBase::getLogger()` [[line 461](#)]

**returns the common logger object (overridden this method in the application class)**

- **Access** protected

*IMailClient* function `ApplicationBase::getMailClient()` [[line 483](#)]

**returns the common mail client object (overridden this method in the application class)**

- **Access** protected

*void* function `ApplicationBase::getPluginHooks()` [[line 291](#)]

**get plugin hooks**

- **Access** public

*TranslatorBase* function *ApplicationBase::getTranslator()* [*line 472*]

**returns the common translator object (overridden this method in the application class)**

- **Access** protected

*void* function *ApplicationBase::handleError(\$errno, \$errstr, \$errfile, \$errline)* [*line 442*]

**Function Parameters:**

- *string* **\$errno** error code
- *string* **\$errstr** error description
- *string* **\$errfile** file
- *string* **\$errline** line no.

**event triggered by an error in the application, can be overridden to provide error handling.**

- **Abstract Element**
- **Access** protected

*void* function *ApplicationBase::handleException(\Exception \$e, \$e)* [*line 429*]

**Function Parameters:**

- *\Exception* **\$e**
- *Exception* **\\$e**

**event triggered by an uncaught Exception thrown in the application, can be overridden**

to provide error handling.

- **Abstract Element**
- **Access** protected

*void function* ApplicationBase::loadAppConfig(\$xmlConfig, \$xmlconfig) [*line* [327](#)]

**Function Parameters:**

- *string* **\$xmlconfig** path to xml config file
- **\$xmlConfig**

**loads config data from an xml file**

- **Access** public

*void function* ApplicationBase::run() [*line* [221](#)]

**begin processing of the servlet**

- **Access** public

## Class AppMessage

[*line* [17](#)]

**Represents a single application message**



- **Package** PHPRum
- **Author** Darnell Shinbine

Constructor *void* function AppMessage::\_\_construct(\$msg, [\$type = null]) [*line* [59](#)]

**Function Parameters:**

- *string* **\$msg** Message content
- [AppMessageType](#) **\$type** Message type as constant of AppMessageType::Success(), AppMessageType::Fail() or AppMessageType::Notice()

## Constructor

- **Access** public

# Class AppMessageType

[*line* [17](#)]

**Represents an application message type**

- **Package** PHPRum
- **Author** Darnell Shinbine

*AppMessageType* function AppMessageType::Fail() [*line* [42](#)]

## Specifies the message type as Fail

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

*AppMessageType* function *AppMessageType::Info()* [*line [30](#)*]

## Specifies the message type as Notice

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

*AppMessageType* function *AppMessageType::Success()* [*line [36](#)*]

## Specifies the message type as Success

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

*AppMessageType* function *AppMessageType::Warning()* [*line [48](#)*]

## Specifies the message type as Notice

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

*string* function *AppMessageType::\_\_toString()* [*line [55](#)*]

Returns a string representing the type

- **Access** public

## Class AppState

[line [17](#)]

Represents an application state

- **Package** PHPRum
- **Author** Darnell Shinbine

*AppState* function AppState::Debug() [line [36](#)]

**Specifies the application state as Debug**

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

*AppState* function AppState::On() [line [30](#)]

**Specifies the application state as On**

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

## Class ArgumentOutOfRangeException

[line [17](#)]

**Represents Argument Out Of Range Exception**

- **Package** PHPRum
- **Author** Darnell Shinbine

## Class BadMemberCallException

[line [17](#)]

**Represents a Bad Member Call Exception**

- **Package** PHPRum
- **Author** Darnell Shinbine

## Class BadMethodCallException

[line [17](#)]

**Represents a Bad Method Call Exception**

- **Package** PHPRum
- **Author** Darnell Shinbine

## Class ConsoleApplicationBase

*[line [19](#)]*

**Provides the base functionality of a console application. This class receives command line data and executes the application**

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Abstract Element**

## Class EventBase

*[line [19](#)]*

**Provides basic event handling**

- **Package** PHPRum
- **Property** string \$name: event name
- **Author** Darnell Shinbine

- **Abstract Element**

Constructor *void* function EventBase::\_\_construct(\$name) [[line 34](#)]

**Function Parameters:**

- *string* **\$name** event name

## Constructor

- **Access** protected

# Class EventHandlerBase

[[line 20](#)]

**Provides basic event handling**

- **Package** PHPRum
- **Property** string \$event: event
- **Property** string \$callback: callback
- **Author** Darnell Shinbine
- **Abstract Element**

Constructor *void* function EventHandlerBase::\_\_construct(\$event, \$callback) [[line 43](#)]

**Function Parameters:**

- *string* **\$event** event

- *string* **\$callback** call back

## Constructor

- **Access** protected

*void* function EventHandlerBase::raise(&\$sender, [\$args = array()], \$sender) [*line* [79](#)]

### **Function Parameters:**

- *object* **\$sender** Sender object
- *array* **\$args** optional args
- **&\$sender**

## raise event

- **Access** public

# Class ExceptionBase

[*line* [18](#)]

**Represents a base Exception**

- **Package** PHPRum

- **Author** Darnell Shinbine
- **Abstract Element**

## Class GridViewChangeEvent

[line [18](#)]

**Provides event handling**

- **Package** PHPRum
- **Author** Darnell Shinbine

Constructor *void* function GridViewChangeEvent::\_\_construct() [line [25](#)]

**Constructor**

- **Access** public

## Class GridViewChangeOrderEventHandler

[line [18](#)]

**Provides event handling**

- **Package** PHPRum



- **Author** Darnell Shinbine

Constructor *void* function GridViewChangeOrderEventHandler::\_\_construct(\$callback) [[line 26](#)]

**Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class GridViewFilterEvent

[[line 18](#)]

## Provides event handling

- **Package** PHPRum
- **Author** Darnell Shinbine

Constructor *void* function GridViewFilterEvent::\_\_construct() [[line 25](#)]

## Constructor

- **Access** public

# Class GridViewFilterEventHandler

[line [18](#)]

**Provides event handling**

- **Package** PHPRum
- **Author** Darnell Shinbine

Constructor *void* function GridViewFilterEventHandler::\_\_construct(\$callback) [line [26](#)]

**Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class GridViewSortEvent

[line [18](#)]

**Provides event handling**

- **Package** PHPRum
- **Author** Darnell Shinbine

Constructor *void* function GridViewSortEvent::\_\_construct() [*line* [25](#)]

## Constructor

- **Access** public

# Class GridViewSortEventHandler

[*line* [18](#)]

**Provides event handling**

- **Package** PHPRum
- **Author** Darnell Shinbine

Constructor *void* function GridViewSortEventHandler::\_\_construct(\$callback) [*line* [26](#)]

## **Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

## Class IndexOutOfRangeException

[line [17](#)]

**Represents an Index Out Of Range Exception**

- **Package** PHPRum
- **Author** Darnell Shinbine

## Class InvalidArgumentException

[line [17](#)]

**Represents an Invalid Argument Exception**

- **Package** PHPRum
- **Author** Darnell Shinbine

## Class InvalidOperationException

[line [17](#)]

**Represents an Invalid Operation Exception**

- **Package** PHPRum
- **Author** Darnell Shinbine

## Class LoggerBase

[[line 17](#)]

**Provides access to read/write log messages**

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Abstract Element**

*void* function LoggerBase::flush(\$category) [[line 44](#)]

**Function Parameters:**

- *string* **\$category** message category

**This method flushes all messages**

- **Abstract Element**
- **Access** public

*void* function LoggerBase::log(\$message, \$category) [[line 26](#)]

#### ***Function Parameters:***

- *string* **\$message** message to log
- *string* **\$category** message category

**This method writes a log message to memory**

- **Abstract Element**
- **Access** public

*array* function LoggerBase::logs(\$category) [*line* [35](#)]

#### ***Function Parameters:***

- *string* **\$category** message category

**This method retrieves all log messages**

- **Abstract Element**
- **Access** public

**Class Make**

[*line* [18](#)]

**deplomyent script**

- **Package** PHPRum
- **Author** Darnell Shinbine

## Make::\$namespace

*string* = " [\[line 24\]](#)

## namespace

- **Static**

*void* function Make::execute(\$argc, \$argv) [\[line 34\]](#)

### **Function Parameters:**

- *int* **\$argc** Number of command line arguments
- *array* **\$argv** Array of command line arguments

## execute the application

- **Access** protected

*array* function Make::getOptions(\$argc, \$argv) [\[line 87\]](#)

### **Function Parameters:**

- *int* **\$argc** Number of command line arguments
- *array* **\$argv** Array of command line arguments

## retrieve command line options

- **Access** protected

void function Make::handleError(\$errno, \$errstr, \$errfile, \$errline) [[line 134](#)]

**Function Parameters:**

- *string* **\$errno** error code
- *string* **\$errstr** error description
- *string* **\$errfile** file
- *string* **\$errline** line no.

event triggered by an error in the application, can be overridden to provide error handling.

- **Access** protected

void function Make::handleException(\Exception \$e, \$e) [[line 121](#)]

**Function Parameters:**

- *\Exception* **\$e**
- *Exception* **\\$e**

event triggered by an uncaught Exception thrown in the application, can be overridden to provide error handling.

- **Access** protected



# Class MethodNotImplementedException

[line [17](#)]

Represents a Method Not Implemented Exception

- **Package** PHPRum
- **Author** Darnell Shinbine

## MethodNotImplementedException::\$message

*mixed* = 'Method Not Implemented' [line [19](#)]

- **Access** protected

# Class MissingArgumentException

[line [17](#)]

Represents a Missing Arguments Exception

- **Package** PHPRum
- **Author** Darnell Shinbine

# Class ModelBase

[[line 17](#)]

This class represents a data model.

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Abstract Element**

*ActiveRecordBase* function ModelBase::getClass(\$args) [[line 60](#)]

**Function Parameters:**

- *array* **\$args** optional associative array of initial properties

**static method to create new ActiveRecordBase of this type**

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

*array* function ModelBase::toArray() [[line 51](#)]

**converts the model into an array**

- **Abstract Element**
- **Access** public

# Class Rum

[[line 20](#)]

## Provides access to application services

It contains such things as the session data, configuration data, server messages and application state. It also provides methods for forwarding, message handling, URL rewriting and logging.

- **Package** PHPRum
- **Author** Darnell Shinbine

*WebApplicationBase* function Rum::app() [[line 27](#)]

## This method returns the application

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

*string* function Rum::baseURI() [[line 38](#)]

## returns the application base uri

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

*string* function Rum::breadcrumb() [[line 49](#)]

**returns the current page breadcrumb trail as an HTML element**

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

*CacheBase* function Rum::cache() [[line 92](#)]

**This method returns the configuration object**

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

*void* function Rum::clearMessages() [[line 188](#)]

**this method removes any displayed messages**

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

*AppConfiguration* function Rum::config() [[line 103](#)]

**This method returns the configuration object**

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

*DataAdapter* function Rum::db() [[line 114](#)]

**This method returns the application DataAdapter object**

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

*string* function Rum::escape(\$string, [\$quote\_style = \_\_QUOTE\_STYLE\_\_], [\$charset = ""]) [[line 128](#)]

**Function Parameters:**

- *string* **\$string** string to escape
- *int* **\$quote\_style** quote style
- *string* **\$charset** character set

**This method returns an escaped string using the application charset for outputting**

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

*void* function Rum::flash(\$msg, [\\\$type = null], \$type) [[line 157](#)]

**Function Parameters:**

- *string* **\$msg** Message content
- [AppMessageType](#) **\$type** Message type as constant of AppMessageType::Success(), AppMessageType::Fail() or AppMessageType::Notice()
- [AppMessageType](#) **\\\$type**

**this method adds message to message stack**

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

*void* function Rum::forward([\$nextPage = "], [\$args = array()], [\\\$method = null], \$method) [*line* [144](#)]

**Function Parameters:**

- *string* **\$nextPage** Name of requested page
- *array* **\$args** args for next action
- [ForwardMethodType](#) **\$method** forward method as constant of ForwardMethodType::URI() or ForwardMethodType::Request()
- [ForwardMethodType](#) **\\\$method**

**This method sets the next URI once the current controller is finished executing, the servlet will re-request the page with the requested action. This method allows you to replace post data with keep friendly urls that can be bookmarked.**

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

*string* function Rum::link(\$title, \$page, [\$args = array()], [\$class = ""]) [*line* [205](#)]

**Function Parameters:**

- *string* **\$title** title of link
- *string* **\$page** name of page
- *array* **\$args** array of args
- *string* **\$class** css class

returns an HTML anchor element

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

*void* function Rum::log(\$message, [\$category = 'events']) [*line* [220](#)]

**Function Parameters:**

- *string* **\$message** event to log
- *string* **\$category** log category

**This method automatically writes a string to a log stamped with the current time**

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

*string* function Rum::messages() [*line* [231](#)]

**returns the application messages as an HTML element**

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

*PageControllerBase* function Rum::requestHandler() [*line* [272](#)]

**This method returns the current controller**

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

*void* function Rum::sendHTTPError([\$statuscode = 500]) [*line* [322](#)]

***Function Parameters:***

- *int* **\$statuscode** HTTP status code

**This method sends an HTTP status message to the client**

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

*string* function Rum::tl(\$stringId, [\$default = ""]) [*line* [336](#)]

***Function Parameters:***

- *string* **\$stringId** string id to translate
- *string* **\$default** string if not found

**This method translates a message into the appropriate language**

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

*void* function Rum::trace(\$var) [*line* [284](#)]

***Function Parameters:***



- *mixed* **\$var** raw variable to trace

**This method traces a variable**

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

*string* function Rum::uri([\$page = "], [\$args = array()]) [*line* [297](#)]

**Function Parameters:**

- *string* **\$page** name of page
- *array* **\$args** array of parameters

**This method returns a URI based on the requested page and parameters**

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

*string* function Rum::url([\$page = "], [\$args = array()]) [*line* [310](#)]

**Function Parameters:**

- *string* **\$page** name of the page
- *array* **\$args** array of parameters

**This method returns a URL based on the requested page and parameters**

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

## Class Tester

[line [18](#)]

**Provides the base functionality of the application. This class receives command line data and executes the application**

- **Package** PHPRum
- **Author** Darnell Shinbine

array function Tester::getAllFunctionalTestCases([\$path = ""]) [line [172](#)]

**Function Parameters:**

- *string* **\$path** initial path

**return all functional test cases**

- **Access** public

array function Tester::getAllUnitTestCases() [line [150](#)]

**return all unit test cases**

- **Access** public

*ControllerTestCaseBase* function `Tester::getFunctionalTestCase($module)` [[line 106](#)]

**Function Parameters:**

- *string* **\$module** name of module

**return functional test case**

- **Access** public

*UnitTestCaseBase* function `Tester::getUnitTestCase($testCase)` [[line 77](#)]

**Function Parameters:**

- *string* **\$testCase** name of test

**return unit test case**

- **Access** public

*void* function `Tester::loadTestConfig($reporter)` [[line 206](#)]

**Function Parameters:**

- *\SimpleReporter* **\$reporter** instance of a *\SimpleReporter*

**run all test cases**

- **Access** public

*void function* Tester::runAllTestCases(\\$reporter, \$reporter) [*line* [52](#)]

***Function Parameters:***

- *\SimpleReporter* **\$reporter** instance of a *\SimpleReporter*
- *SimpleReporter* **\\$reporter**

## run all test cases

- **Access** public

*void function* Tester::runFunctionalTestCase(\$testCase, \\$reporter, \$reporter) [*line* [40](#)]

***Function Parameters:***

- *string* **\$testCase** name of module
- *\SimpleReporter* **\$reporter** instance of a *\SimpleReporter*
- *SimpleReporter* **\\$reporter**

## run functional test case

- **Access** public

*void function* Tester::runUnitTestCase(\$testCase, \\$reporter, \$reporter) [*line* [27](#)]

***Function Parameters:***

- *string* **\$testCase** name of test

- `\SimpleReporter $reporter` instance of a `\SimpleReporter`
- `SimpleReporter $reporter`

run unit test case

- **Access** public

## Class TypeMismatchException

[line [17](#)]

Represents a Type Mismatch Exception

- **Package** PHPRum
- **Author** Darnell Shinbine

## Class WebApplicationBase

[line [31](#)]

This class represents the base web application. This class receives HTTP request input and delegates the request to a Controller. Once the Controller has handled the request, this class will regain control and render the appropriate View selected by the Controller unless there are additional actions to perform.

- **Package** PHPRum
- **Property** [PageControllerBase](#) \$requestHandler: Reference to the current ControllerBase
- **Property** string \$prevPage: Previous page
- **Property** string \$currentPage: Current page
- **Property** string \$forwardPage: Page to forward to
- **Property** string \$forwardURI: URI to forward to
- **Property** array \$forwardParams: array of forward parameters
- **Property** [Session](#) \$session: Contains the Session object
- **Property** [HttpRequest](#) \$request: Contains the HttpRequest object
- **Property** [AppMessageCollection](#) \$messages: Contains the AppMessageCollection object
- **Author** Darnell Shinbine
- **Abstract Element**

Constructor *void* function `WebApplicationBase::__construct()` [*line 107*]

### **Constructor**

Creates an instance of the controller and sets default action map.

- **Access** public

*void* function `WebApplicationBase::clearForwardPage()` [*line 375*]

### **This method clear all forward URI's**

- **Access** public

*void* function `WebApplicationBase::execute()` [*line 490*]

### **execute the application**

- **Access** protected

array function WebApplicationBase::getAllControllers([\$path = "]) [[line 400](#)]

**Function Parameters:**

- *string* **\$path** initial path

**return all controllers**

- **Access** public

string function WebApplicationBase::getPageURI([\$page = "], [\$args = array()]) [[line 228](#)]

**Function Parameters:**

- *string* **\$page** name of page
- *array* **\$args** array of parameters

**returns a URI based on the requested page and parameters**

- **Access** public

ControllerBase function WebApplicationBase::getRequestHandler(\$controller) [[line 183](#)]

**Function Parameters:**

- *string* **\$controller** name of the controller

**return the requestHandler**

- **Access** public

*Translator* function `WebApplicationBase::getSession()` [[line 479](#)]

**returns the common translator object (overridden this method in the application class)**

- **Access** protected

*void* function `WebApplicationBase::handleError($errno, $errstr, $errfile, $errline)` [[line 831](#)]

**Function Parameters:**

- *string* **\$errno** error code
- *string* **\$errstr** error description
- *string* **\$errfile** file
- *string* **\$errline** line no.

**event triggered by an error in the application, can be overridden to provide error handling.**

- **Access** protected

*void* function `WebApplicationBase::handleException(\Exception $e)` [[line 692](#)]

**Function Parameters:**



- *Exception* \$e
- *Exception* \ \$e

event triggered by an uncaught Exception thrown in the application, can be overridden to provide error handling.

- **Access** protected

*void function* WebApplicationBase::loadApplicationState() [*line* [433](#)]

**This method retrieves an instance of the previous action and the servlet messages. These can then be used by the servlet to populate the view.**

- **Access** public

*void function* WebApplicationBase::requestProcessor(\\&\$request, \$request) [*line* [528](#)]

**Function Parameters:**

- [HttpRequest](#) \$request HttpRequest object
- [HttpRequest](#) \\&\$request

**Processe HTTP Request and create actions to handle business logic. Then render the view.**

Can be overridden to provide additional or alternative private functionality.

- **Access** protected

`void function WebApplicationBase::saveApplicationState()` [[line 461](#)]

**This method stores an instance of the previous action and the servlet messages. These can then be used by the servlet after a page request to populate the view.**

- **Access** public

`void function WebApplicationBase::sendHTTPError([$statusCode = 500])` [[line 313](#)]

**Function Parameters:**

- *int* **\$statusCode** HTTP status code

**send HTTP status message to client**

- **Access** public

`void function WebApplicationBase::setForwardPage([$nextPage = "], [$args = array()], [$method = null])` [[line 346](#)]

**Function Parameters:**

- *string* **\$nextPage** Name of requested page
- *array* **\$args** args for next action
- [ForwardMethodType](#) **\$method** forward method as constant of `ForwardMethodType::URI()` or `ForwardMethodType::Request()`

**This method sets the next URI once the current controller is finished executing, the servlet will re-request the page with the requested action. This method allows you to replace post data with keep friendly urls that can be bookmarked.**

- **Access** public

`void function WebApplicationBase::trace($var) [line 388]`

***Function Parameters:***

- **\$var**

## Trace a variable

- **Access** public

# activerecordbase.class.php

- **Package** PHPRum
- **Author** Darnell Shimbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# activerecordcollection.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# activerecordexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# relationshiptype.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class ActiveRecordBase

[line 32]

**This class represents a table withing a database or an instance of a single record in a table and provides database abstraction.**

The ActiveRecordBase exposes 5 protected properties, do not define these properties in the sub class to have the properties auto determined

- **Package** PHPRum
  - **Sub-Package** ActiveRecord
  - **Property** string \$table: Specifies the table name
  - **Property** string \$pkey: Specifies the primary key (there can only be one primary key defined)
  - **Property** array \$fields: Specifies field names mapped to field types
  - **Property** array \$rules: Specifies field names mapped to field rules
  - **Property** array \$relationships: Specifies table relationships
- 2 Additional public readonly properties are exposed
- **Property** bool \$isNull: Specifies whether ActiveRecord is null

- **Property** bool \$isEmpty: Specifies whether ActiveRecord is empty
- **Author** Darnell Shinbine
- **Abstract Element**

#### ActiveRecordBase::\$pkey

*string* = " [[line 44](#)]

**Specifies the primary key (there can only be one primary key defined)**

- **Access** protected

#### ActiveRecordBase::\$relationships

*array* = array() [[line 50](#)]

**Specifies table relationships**

- **Access** protected

#### ActiveRecordBase::\$table

*string* = " [[line 38](#)]

**Specifies the table name**

- **Access** protected

Constructor *void* function ActiveRecordBase::\_\_construct() [[line 64](#)]

**Constructor**



- **Access** protected

*void* function ActiveRecordBase::afterCreate() [[line 847](#)]

**event called after a new ActiveRecordBase is created**

- **Access** protected

*void* function ActiveRecordBase::afterDelete() [[line 927](#)]

**event called after every delete**

- **Access** protected

*void* function ActiveRecordBase::afterInsert() [[line 895](#)]

**event called after every insert**

- **Access** protected

*void* function ActiveRecordBase::afterRetrieve() [[line 863](#)]

**event called after every retrieve**

- **Access** protected

*void* function ActiveRecordBase::afterSave() [[line 879](#)]

**event called after every save**

- **Access** protected

*void* function ActiveRecordBase::afterUpdate() [[line 911](#)]

**event called after every update**

- **Access** protected

*DataSet* function ActiveRecordBase::all([\$args = array()]) [[line 628](#)]

**Function Parameters:**

- *array* **\$args** associative array of keys and values

**static method to return a DataSet of this type**

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

*void* function ActiveRecordBase::beforeCreate() [[line 839](#)]

**event called before a new ActiveRecordBase is created**

- **Access** protected

*void* function ActiveRecordBase::beforeDelete() [[line 919](#)]  
**event called before every delete**

- **Access** protected

*void* function ActiveRecordBase::beforeInsert() [[line 887](#)]  
**event called before every insert**

- **Access** protected

*void* function ActiveRecordBase::beforeRetrieve() [[line 855](#)]  
**event called before every retrieve**

- **Access** protected

*void* function ActiveRecordBase::beforeSave() [[line 871](#)]  
**event called before every save**

- **Access** protected

*void* function ActiveRecordBase::beforeUpdate() [[line 903](#)]  
**event called before every update**

- **Access** protected

*DataSet* function ActiveRecordBase::count([\$args = array()]) [*line* [652](#)]

**Function Parameters:**

- *array* **\$args** associative array of keys and values

**static method to return a count of records of this type**

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

*ActiveRecordBase* function ActiveRecordBase::create([\$args = array()]) [*line* [553](#)]

**Function Parameters:**

- *array* **\$args** optional associative array of initial properties

**static method to create new ActiveRecordBase of this type**

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

*void* function ActiveRecordBase::delete() [*line* [471](#)]

**delete ActiveRecordBase**

- **Access** public

*void* function ActiveRecordBase::deleteAll() [*line* [492](#)]

**delete ActiveRecordBase and all associated ActiveRecordBases (cascading delete)**

- **Access** public

*ActiveRecordBase* function ActiveRecordBase::find(\$args) [*line* [565](#)]

**Function Parameters:**

- *array* **\$args** associative array of keys and values

**static method to find ActiveRecordBase of this type**

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

*ActiveRecordCollection* function ActiveRecordBase::findAll([\$args = array()]) [*line* [592](#)]

**Function Parameters:**

- *array* **\$args** associative array of keys and values

**static method to find ActiveRecordBases of this type**

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

*ActiveRecordBase* function *ActiveRecordBase::findById(\$pid)* [[line 577](#)]

**Function Parameters:**

- *mixed* **\$pid** key value to lookup

**static method to find ActiveRecordBase by primary key**

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

*ActiveRecordBase* function *ActiveRecordBase::first([\$args = array()])* [[line 604](#)]

**Function Parameters:**

- *string* **\$args** associative array of keys and values

**static method to find the first ActiveRecordBase of this type**

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

*Form* function *ActiveRecordBase::form(\$controlId)* [[line 664](#)]

**Function Parameters:**

- *string* **\$controlId** form id

**static method to return a Form object**

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

*ActiveRecordBase* function *ActiveRecordBase::last*([\$args = array()]) [*line* [616](#)]

**Function Parameters:**

- *string* **\$args** associative array of keys and values

**static method to find the last ActiveRecordBase of this type**

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

*DataSet* function *ActiveRecordBase::more*([\$args = array()]) [*line* [640](#)]

**Function Parameters:**

- *array* **\$args** associative array of keys and values

**static method to return an extended DataSet of this type**

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

*void* function *ActiveRecordBase::save*() [*line* [445](#)]

**update DataAdapter**

- **Access** public

array function ActiveRecordBase::toArray() [[line 536](#)]  
**returns an array containing all row data**

- **Access** public

## Class ActiveRecordCollection

[[line 19](#)]

**Represents a collection of ActiveRecord**

- **Package** PHPRum
- **Sub-Package** ActiveRecord
- **Author** Darnell Shinbine

void function ActiveRecordCollection::add(\$item) [[line 51](#)]

**Function Parameters:**

- [ActiveRecordBase](#) \$item

**Add an ActiveRecord to the Collection**



- **Access** public

*ActiveRecordBase* function *ActiveRecordCollection::itemAt(\$index)* [*line* [70](#)]

**Function Parameters:**

- *int* **\$index** index of ActiveRecord

**return ActiveRecord at a specified index**

- **Access** public

## Class ActiveRecordException

[*line* [18](#)]

**Represents a Bad Member Call Exception**

- **Package** PHPRum
- **Sub-Package** ActiveRecord
- **Author** Darnell Shinbine

## Class RelationshipType

[line [18](#)]

## Represents a relationship type

- **Package** PHPRum
- **Sub-Package** ActiveRecord
- **Author** Darnell Shinbine

*RelationshipType* function RelationshipType::BelongsTo() [line [37](#)]

### **Specifies the relationship is a many to one**

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

*RelationshipType* function RelationshipType::HasMany() [line [31](#)]

### **Specifies the relationship is a one to many**

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

*RelationshipType* function RelationshipType::HasManyAndBelongsTo() [line [43](#)]

### **Specifies the relationship is a many to many**

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

*string* function RelationshipType::\_\_toString() [*line 49*]

**Returns a string representing the type**

- **Access** public

# applicationrunevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# applicationruneventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# pagecontrollercreatepageevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# pagecontrollercreatepageeventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# webapplicationhandlerequestevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# webapplicationhandlerequesteventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# webapplicationloadstateevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# webapplicationloadstateeventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# webapplicationsavestateevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# webapplicationsavestateeventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class ApplicationRunEvent

[line [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Base
- **Author** Darnell Shinbine

Constructor *void* function ApplicationRunEvent::\_\_construct() [line [26](#)]

### **Constructor**

- **Access** public

# Class ApplicationRunEventHandler

[line [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Base
- **Author** Darnell Shinbine

Constructor *void* function ApplicationRunEventHandler::\_\_construct(\$callback) [line [27](#)]

**Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class PageControllerCreatePageEvent

[line [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Base

- **Author** Darnell Shinbine

Constructor *void* function PageControllerCreatePageEvent::\_\_construct() [[line 26](#)]

## Constructor

- **Access** public

# Class PageControllerCreatePageEventHandler

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Base
- **Author** Darnell Shinbine

Constructor *void* function PageControllerCreatePageEventHandler::\_\_construct(\$callback) [[line 27](#)]

## Function Parameters:

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class WebApplicationHandleRequestEvent

[*line* [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Base
- **Author** Darnell Shinbine

Constructor *void* function WebApplicationHandleRequestEvent::\_\_construct() [*line* [26](#)]

## **Constructor**

- **Access** public

# Class WebApplicationHandleRequestEventHandler

[*line* [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Base



- **Author** Darnell Shinbine

Constructor *void* function WebApplicationHandleRequestEventHandler::\_\_construct(\$callback) [*line 27*]

**Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class WebApplicationLoadStateEvent

[*line 19*]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Base
- **Author** Darnell Shinbine

Constructor *void* function WebApplicationLoadStateEvent::\_\_construct() [*line 26*]

## Constructor

- **Access** public

# Class WebApplicationLoadStateEventHandler

[line [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Base
- **Author** Darnell Shinbine

Constructor *void* function WebApplicationLoadStateEventHandler::\_\_construct(\$callback) [line [27](#)]

**Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class WebApplicationSaveStateEvent

[line [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Base
- **Author** Darnell Shinbine

Constructor *void* function WebApplicationSaveStateEvent::\_\_construct() [[line 26](#)]

### **Constructor**

- **Access** public

## Class WebApplicationSaveStateEventHandler

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Base
- **Author** Darnell Shinbine

Constructor *void* function WebApplicationSaveStateEventHandler::\_\_construct(\$callback) [[line 27](#)]

### **Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# build.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# cachebase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# cacheexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# dbcache.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# filecache.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# memcache.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class Build

[line [18](#)]

**Provides access to read/write build files**

- **Package** PHPRum
- **Sub-Package** Caching
- **Author** Darnell Shinbine

**Build::\$verbose**

*bool* = false [line [24](#)]

**verbose**

- **Static**

*void* function Build::build() [[line 155](#)]

### **Rebuild build folder**

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

*void* function Build::clean() [[line 128](#)]

### **Cleans and prepares build folder**

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

*void* function Build::clear(\$id) [[line 103](#)]

#### ***Function Parameters:***

- *string* **\$id** the key identifying the value to be cached

### **Deletes a value with the specified key from cache**

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

*string* function Build::get(\$id, \$expire) [[line 35](#)]

#### ***Function Parameters:***

- *string* **\$id** the key identifying the value to be cached
- *integer* **\$expire** the expiration time of the value in seconds

1. means never expire,

**Retrieves a value from cache with a specified key.**

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

*array* function Build::getFileArray() [*line* [185](#)]

**get id array**

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

*void* function Build::put(\$id, \$value) [*line* [68](#)]

**Function Parameters:**

- *string* **\$id** the key identifying the value to be cached
- *string* **\$value** the value to be cached

**Stores a value identified by a key into cache.**

If the cache already contains such a key, the existing value and expiration time will be replaced with the new ones.

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

# Class CacheBase

[[line 18](#)]

**Provides access to read/write cache objects**

- **Package** PHPRum
- **Sub-Package** Caching
- **Author** Darnell Shinbine
- **Abstract Element**

*void* function CacheBase::clear(\$id) [[line 52](#)]

**Function Parameters:**

- *string* **\$id** the key identifying the value to be cached

**Deletes a value with the specified key from cache**

- **Abstract Element**
- **Access** public

*void* function CacheBase::flush() [[line 60](#)]

**Deletes all values from cache.**

- **Abstract Element**
- **Access** public

*string* function CacheBase::get(\$id, \$expire) [*line* [28](#)]

**Function Parameters:**

- *string* **\$id** the key identifying the value to be cached
- *integer* **\$expire** the expiration time of the value in seconds
  1. means never expire,

**Retrieves a value from cache with a specified key.**

- **Abstract Element**
- **Access** public

*void* function CacheBase::put(\$id, \$value, [\$expires = 0]) [*line* [43](#)]

**Function Parameters:**

- *string* **\$id** the key identifying the value to be cached
- *string* **\$value** the value to be cached
- *integer* **\$expires** the expiration time of the value in seconds
  1. means never expire,

**Stores a value identified by a key into cache.**

If the cache already contains such a key, the existing value and expiration time will be replaced with the new ones.

- **Abstract Element**
- **Access** public

## Class CacheException

[line [19](#)]

**Represents a Cache Exception**

- **Package** PHPRum
- **Sub-Package** Caching
- **Author** Darnell Shinbine

## Class DBCache

[line [18](#)]

**Provides access to read/write cache objects from a database**

- **Package** PHPRum
- **Sub-Package** Caching
- **Author** Darnell Shinbine

Constructor *void* function `DBCache::__construct([$table = ''], [$dsn = ''])` [[line 39](#)]

**Function Parameters:**

- *string* **\$table** table name
- *string* **\$dsn** database connection string

## Constructor

- **Access** public

*void* function `DBCache::clear($id)` [[line 146](#)]

**Function Parameters:**

- *string* **\$id** the key identifying the value to be cached

## Deletes a value with the specified key from cache

- **Access** public

*void* function `DBCache::flush()` [[line 161](#)]

**Deletes all values from cache.**

- **Access** public

*string* function `DBCache::get($id)` [[line 63](#)]

**Function Parameters:**



- *string* **\$id** the key identifying the value to be cached

**Retrieves a value from cache with a specified key.**

- **Access** public

array function DBCache::getIdArray() [[line 174](#)]  
**get id array**

- **Access** public

void function DBCache::put(\$id, \$value, [\$expires = 0]) [[line 124](#)]  
**Function Parameters:**

- *string* **\$id** the key identifying the value to be cached
- *string* **\$value** the value to be cached
- *integer* **\$expires** the expiration time of the value in seconds
  1. means never expire,

**Stores a value identified by a key into cache.**

If the cache already contains such a key, the existing value and expiration time will be replaced with the new ones.

- **Access** public

# Class FileCache

[[line 18](#)]

**Provides access to read/write cache files**

- **Package** PHPRum
- **Sub-Package** Caching
- **Author** Darnell Shinbine

*void function FileCache::clear(\$id) [[line 98](#)]*

***Function Parameters:***

- *string* **\$id** the key identifying the value to be cached

**Deletes a value with the specified key from cache**

- **Access** public

*void function FileCache::flush() [[line 121](#)]*

**Deletes all values from cache.**

- **Access** public

*string function FileCache::get(\$id) [[line 26](#)]*

### ***Function Parameters:***

- *string* **\$id** the key identifying the value to be cached

**Retrieves a value from cache with a specified key.**

- **Access** public

*array* function FileCache::getIdArray() [*line* [143](#)]

**get id array**

- **Access** public

*void* function FileCache::put(\$id, \$value, [\$expires = 0]) [*line* [65](#)]

### ***Function Parameters:***

- *string* **\$id** the key identifying the value to be cached
- *string* **\$value** the value to be cached
- *integer* **\$expires** the expiration time of the value in seconds
  1. means never expire,

**Stores a value identified by a key into cache.**

If the cache already contains such a key, the existing value and expiration time will be replaced with the new ones.

- **Access** public

# Class MEMCache

[[line 18](#)]

Provides access to read/write cache objects from memcache

- **Package** PHPRum
- **Sub-Package** Caching
- **Author** Darnell Shinbine

Constructor *void* function MEMCache::\_\_construct([\$server = 'localhost'], [\$port = 11211]) [[line 38](#)]

## **Function Parameters:**

- *string* **\$server** server, default is localhost
- *string* **\$port** port, default is 11211

## **Constructor**

- **Access** public

*void* function MEMCache::clear(\$id) [[line 91](#)]

## **Function Parameters:**

- *string* **\$id** the key identifying the value to be cached

## Deletes a value with the specified key from cache

- **Access** public

*void* function MEMCache::flush() [*line* [102](#)]

**Deletes all values from cache.**

- **Access** public

*string* function MEMCache::get(\$id) [*line* [58](#)]

**Function Parameters:**

- *string* **\$id** the key identifying the value to be cached

**Retrieves a value from cache with a specified key.**

- **Access** public

*void* function MEMCache::put(\$id, \$value, [\$expires = 0], \$expire) [*line* [76](#)]

**Function Parameters:**

- *string* **\$id** the key identifying the value to be cached
- *string* **\$value** the value to be cached
- *integer* **\$expire** the expiration time of the value in seconds
  1. means never expire,

- **\$expires**

**Stores a value identified by a key into cache.**

If the cache already contains such a key, the existing value and expiration time will be replaced with the new ones.

- **Access** public

# appmessagecollection.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# appsettingscollection.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# eventargs.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# eventcollection.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# eventhandlercollection.class.php

- **Package** PHPRum
- **Author** Darnell Shimbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# collectionbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# dictionarybase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# iteratorbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# keycollection.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# stringcollection.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# stringdictionary.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# valuecollection.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class AppMessageCollection

[line [18](#)]

**Represents a collection of Application Messages**

- **Package** PHPRum
- **Sub-Package** Collections
- **Author** Darnell Shinbine

*bool* function AppMessageCollection::add(\$item) [line [62](#)]

**Function Parameters:**

- [AppMessage](#) \$item

**add AppMessage to Collection**

- **Access** public

*StringCollection* function AppMessageCollection::getByType([\$type = null]) [*line* [94](#)]

**Function Parameters:**

- *constant \$type* Message type as constant of AppMessageType::Success(), AppMessageType::Fail() or AppMessageType::Notice()

**return Collection of messages**

- **Access** public

*AppMessage* function AppMessageCollection::itemAt(\$index) [*line* [82](#)]

**Function Parameters:**

- *int \$index* index of ActiveRecord

**return AppMessage at a specified index**

- **Access** public

*array* function AppMessageCollection::toString() [*line* [114](#)]

**convert Iterator object into string**

- **Access** public

# Class AppSettingsCollection

[[line 18](#)]

**Represents a Collection of user-defined settings**

- **Package** PHPRum
- **Sub-Package** Collections
- **Author** Darnell Shinbine

# Class CollectionBase

[[line 18](#)]

**Provides the base functionality for creating Collections**

- **Package** PHPRum
- **Sub-Package** Collections
- **Author** Darnell Shinbine
- **Abstract Element**

Constructor *void* function CollectionBase::\_\_construct([\$collection = null]) [[line 26](#)]

**Function Parameters:**

- *mixed* **\$collection** can be CollectionBase or array used to initialize Collection

## Constructor

- **Access** public

*void* function CollectionBase::add(\$item) [*line* [70](#)]

**Function Parameters:**

- *mixed* **\$item** item

## add item to Collection

- **Access** public

*bool* function CollectionBase::contains(\$item) [*line* [82](#)]

**Function Parameters:**

- *mixed* **\$item** item

## returns true if item array contains item

- **Access** public

*int* function CollectionBase::indexOf(\$item) [*line* [95](#)]

**Function Parameters:**

- *mixed* **\$item** item

**returns first index of item found in Collection**

- **Access** public

*mixed* function CollectionBase::itemAt(\$index) [*line* [115](#)]

**Function Parameters:**

- *int* **\$index** index of item

**return item at a specified index**

- **Access** public

*bool* function CollectionBase::remove(\$item) [*line* [135](#)]

**Function Parameters:**

- *mixed* **\$item** item

**remove item from array**

- **Access** public

*void* function CollectionBase::removeAll() [*line* [176](#)]

**emptys the collection**

- **Access** public

*void* function `CollectionBase::removeAt($index)` [[line 157](#)]

**Function Parameters:**

- *int* **\$index** index to remove at

**remove item from array at a specified index**

- **Access** public

## Class DictionaryBase

[[line 21](#)]

**Provides the base functionality for creating Dictionaries**

- **Package** PHPRum
- **Sub-Package** Collections
- **Property** [KeyCollection](#) \$keys: Collection of keys
- **Property** [ValueCollection](#) \$values: Collection of values
- **Author** Darnell Shinbine
- **Abstract Element**

Constructor *void* function `DictionaryBase::__construct([$dictionary = null])` [[line 29](#)]

**Function Parameters:**

- *mixed* **\$dictionary** can be DictionaryBase or array used to initialize Dictionary

## Constructor

- **Access** public

*void* function DictionaryBase::add(\$key, \$value) [*line* [80](#)]

### **Function Parameters:**

- *mixed* **\$key** key
- *mixed* **\$value** value

**Adds a new key/item pair to a Dictionary object if key does not already exist**

- **Access** public

*bool* function DictionaryBase::contains(\$key) [*line* [118](#)]

### **Function Parameters:**

- *mixed* **\$key** key

**returns a Boolean value that indicates whether a specified key exists in the Dictionary object**

- **Access** public



*int* function DictionaryBase::indexOf(\$key) [*line* [130](#)]

**Function Parameters:**

- *mixed* **\$key** key

**returns index of key found in Dictionary**

- **Access** public

*mixed* function DictionaryBase::item(\$key) [*line* [99](#)]

**Function Parameters:**

- *mixed* **\$key** key

**return the value of a key in the Dictionary**

- **Access** public

*mixed* function DictionaryBase::itemAt(\$index) [*line* [150](#)]

**Function Parameters:**

- *int* **\$index** index

**returns the value of a specified index**

- **Access** public

*mixed* function DictionaryBase::keyAt(\$index) [*line* [170](#)]

**Function Parameters:**

- *int* **\$index** index

**returns the key of a specified index**

- **Access** public

*void* function DictionaryBase::remove(\$key) [*line* [190](#)]

**Function Parameters:**

- *string* **\$key** key to remove

**remove key/value pair from Dictionary**

- **Access** public

*void* function DictionaryBase::removeAll() [*line* [221](#)]

**emptys the dictionary**

- **Access** public

*void* function DictionaryBase::removeAt(\$index) [*line* [210](#)]

### ***Function Parameters:***

- ***int \$index*** index to remove

**remove key/value of a specified index**

- **Access** public

## **Class EventArgs**

*[line [18](#)]*

**Represents a Collection of Event args**

- **Package** PHPRum
- **Sub-Package** Collections
- **Author** Darnell Shinbine

## **Class EventCollection**

*[line [21](#)]*

**Represents a Collection of EventHandlerBase objects**

- **Package** PHPRum
- **Sub-Package** Collections
- **Property** [EventHandlerCollection](#) \$eventHandlers: event handler collection
- **Property** [EventCollection](#) \$events: event collection
- **Author** Darnell Shinbine

Constructor *void* function EventCollection::\_\_construct([\$collection = null]) [*line 36*]

**Function Parameters:**

- *mixed* **\$collection** can be CollectionBase or array used to initialize Collection

## Constructor

- **Access** public

*bool* function EventCollection::contains(\$item) [*line 119*]

**Function Parameters:**

- [EventBase](#) **\$item** item

**returns true if item array contains item**

- **Access** public

*int* function EventCollection::indexOf(\$item) [*line 144*]

**Function Parameters:**

- [EventBase](#) **\$item** item

**returns first index of item found in Collection**

- **Access** public

*void* function EventCollection::raise(\$event, &\$sender, [\$args = array()], \$sender) [*line* [185](#)]

**Function Parameters:**

- [EventBase](#) **\$event** event
- *object* **\$sender** Sender object
- *array* **\$args** optional args
- **&\$sender**

**raise all events**

- **Access** public

*void* function EventCollection::registerEventHandler(\$eventHandler) [*line* [171](#)]

**Function Parameters:**

- [EventHandlerBase](#) **\$eventHandler**

**register event handler**

- **Access** public

# Class EventHandlerCollection

[[line 18](#)]

Represents a Collection of EventHandlerBase objects

- **Package** PHPRum
- **Sub-Package** Collections
- **Author** Darnell Shinbine

*void function* EventHandlerCollection::raise(\$event, &\$sender, [\$args = array()], \$sender) [[line 86](#)]

**Function Parameters:**

- [EventBase](#) **\$event** event
- *object* **\$sender** Sender object
- *array* **\$args** optional args
- **&\$sender**

**raise all events**

- **Access** public

*void function* EventHandlerCollection::registerEventHandler(\$eventHandler) [[line 72](#)]

**Function Parameters:**

- [EventHandlerBase](#) **\$eventHandler**

## register event handler

- **Access** public

# Class IteratorBase

*[line [23](#)]*

**Provides the base functionality for creating Iterators**

- **Package** PHPRum
- **Sub-Package** Collections
- **Property** int \$count: number of collection items
- **Author** Darnell Shinbine
- **Abstract Element**

## IteratorBase::\$items

*array = array() [line [29](#)]*

## internal array

- **Access** protected

Constructor *void* function IteratorBase::\_\_construct([\$iterator = null]) [[line 73](#)]

**Function Parameters:**

- [IteratorBase](#) **\$iterator** used to initialize DictionaryBase

## Constructor

- **Access** public

*array* function IteratorBase::toArray() [[line 201](#)]

**convert Iterator object into array**

- **Access** public

## Class KeyCollection

[[line 18](#)]

**Represents a collection of keys**

- **Package** PHPRum
- **Sub-Package** Collections
- **Author** Darnell Shinbine

*string* function KeyCollection::itemAt(\$index) [[line 26](#)]

**Function Parameters:**



- *int* **\$index** index of item

**return key at a specified index**

- **Access** public

## Class StringCollection

[line [18](#)]

**Represents a collection of strings**

- **Package** PHPRum
- **Sub-Package** Collections
- **Author** Darnell Shinbine

*void* function StringCollection::add(\$item) [line [50](#)]

**Function Parameters:**

- *string* **\$item** item

**add item to Collection**

- **Access** public

*string* function StringCollection::itemAt(\$index) [*line* [69](#)]

**Function Parameters:**

- *int* **\$index** index of item

**return item at a specified index**

- **Access** public

*void* function StringCollection::trim() [*line* [80](#)]

**trim whitespace for all values**

- **Access** public

## Class StringDictionary

[*line* [18](#)]

**Represents a dictionary of key/value strings**

- **Package** PHPRum
- **Sub-Package** Collections
- **Author** Darnell Shinbine

*void* function StringDictionary::add(\$key, \$value) [[line 51](#)]

**Function Parameters:**

- *string* **\$key** key
- *string* **\$value** value

**Adds a new key/item pair to a StringCollection if key does not already exist**

- **Access** public

*string* function StringDictionary::item(\$key) [[line 77](#)]

**Function Parameters:**

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

**return the value of a key in the Dictionary**

- **Access** public

*string* function StringDictionary::itemAt(\$index) [[line 89](#)]

**Function Parameters:**

- *int* **\$index** index

**returns the value of a specified index**

- **Access** public

*string* function StringDictionary::keyAt(\$index) [[line 101](#)]

**Function Parameters:**

- *int* **\$index** index

**returns the key of a specified index**

- **Access** public

*void* function StringDictionary::trim() [[line 112](#)]

**trim whitespace for all values**

- **Access** public

## Class ValueCollection

[[line 18](#)]

**Represents a collection of values**

- **Package** PHPRum
- **Sub-Package** Collections
- **Author** Darnell Shinbine

*string* function ValueCollection::itemAt(\$index) [*line* [26](#)]

***Function Parameters:***

- *int* **\$index** index of item

**return value at a specified index**

- **Access** public

# httpwebrequest.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# httpwebresponse.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class HTTPWebRequest

[line [25](#)]

**Represents an HTTP request (supports SSL)**

- **Package** PHPRum
- **Sub-Package** Comm
- **Property** string \$method: HTTP request method
- **Property** string \$contentType: content-type
- **Property** string \$referer: HTTP referer
- **Property** string \$timeout: timeout
- **Property** string \$httpVersion: HTTP version
- **Property** string \$url: URL to send to
- **Author** Darnell Shinbine

*void* function HTTPWebRequest::addHeader(\$header) [line [173](#)]

**Function Parameters:**

- *string* **\$header** header to set

## add header to request

*SocketStream* function HTTPWebRequest::getRequestStream() [[line 332](#)]

**returns a Stream object for request**

- **Access** protected

*string* function HTTPWebRequest::getRequestString() [[line 254](#)]

**returns a string for request**

- **Access** protected

*HTTPWebResponse* function HTTPWebRequest::getResponse() [[line 235](#)]

**sends request and creates HTTPWebResponse object before closing Stream object**

*void* function HTTPWebRequest::sendRequest() [[line 222](#)]

**sends request and closes Stream object**

*void* function HTTPWebRequest::setData(\$data) [[line 192](#)]

**Function Parameters:**

- *mixed* **\$data** data to set (string|array)

**set data from string or array**



# Class HTTPWebResponse

[[line 23](#)]

Represents the response from a HTTPWebRequest message

- **Package** PHPRum
- **Sub-Package** Comm
- **Property** string \$httpStatus: response headers
- **Property** array \$headers: response headers
- **Property** string \$contentType: response headers
- **Property** string \$content: response headers
- **Author** Darnell Shinbine

*void* function HTTPWebResponse::readFromStream(\\&\$stream, \$header) [[line 88](#)]

**Function Parameters:**

- *string* **\$header** header to set
- [StreamBase](#) \\&\$stream

**read data from Stream object**

- **Access** public

# columncompare.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# columnschema.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# dataadapter.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# dataadapterexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# databaseexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# databaseschema.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# dataset.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# datasettype.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# dirdataadapter.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# mssqldataadapter.class.php

- **Package** PHPRum
- **Author** Tahsin Zulkarnine
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# mssqlquerybuilder.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# mssqltransaction.class.php

- **Package** PHPRum
- **Author** Tahsin Zulkarnine
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# mysqlidataadapter.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# mysqlquerybuilder.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# mysqltransaction.class.php

- **Package** PHPRum
- **Author** Darnell Shimbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# mysqldataadapter.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# mysqlquerybuilder.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# mysqltransaction.class.php

- **Package** PHPRum
- **Author** Darnell Shimbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# pdodataadapter.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# pdoquerybuilder.class.php

- **Package** PHPRum
- **Author** Tahsin Zulkarnine
- **Copyright** Copyright (c) 2013
- **Filesource** [Source Code for this file](#)
- **License** see

# pdotransaction.class.php

- **Package** PHPRum
- **Author** Tahsin Zulkarnine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# querybuilderbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# queryexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# schemaexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# tableschema.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# textdataadapter.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# transactionbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# transactionexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class ColumnCompare

[line [18](#)]

Provides array comparing by type

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Darnell Shinbine

Constructor *void* function ColumnCompare::\_\_construct(\$index, \$field) [line [34](#)]

**Function Parameters:**

- *string* **\$field** column name
- **\$index**

### Constructor

sets column name to compare

- **Access** public

*int* function ColumnCompare::compareDateString(\$a, \$b) [[line 75](#)]

**Function Parameters:**

- *ints* **\$a** ints a
- *ints* **\$b** ints b

**method compares ints and returns -1, 0, 1**

- **Access** public

*int* function ColumnCompare::compareInt(\$a, \$b) [[line 124](#)]

**Function Parameters:**

- *ints* **\$a** ints a
- *ints* **\$b** ints b

**method compares ints and returns -1, 0, 1**

- **Access** public

*int* function ColumnCompare::compareNumeric(\$a, \$b) [[line 104](#)]

**Function Parameters:**

- *ints* **\$a** ints a

- *ints* **\$b** ints b

**method compares numbers and returns -1, 0, 1**

- **Access** public

*int* function ColumnCompare::compareString(\$a, \$b) [[line 60](#)]

**Function Parameters:**

- *string* **\$a** string a
- *string* **\$b** string b

**method compares strings and returns -1, 0, 1**

- **Access** public

*int* function ColumnCompare::compareStringi(\$a, \$b) [[line 45](#)]

**Function Parameters:**

- *string* **\$a** string a
- *string* **\$b** string b

**method compares strings and returns -1, 0, 1**

- **Access** public

# Class ColumnSchema

[line 40]

Represents a column schema

- **Package** PHPRum
- **Sub-Package** DB
- **Property** string \$name: column name
- **Property** string \$table: table name
- **Property** string \$type: column type
- **Property** int \$length: column length
- **Property** bool \$notNull: specifies whether column is not null
- **Property** bool \$primaryKey: specifies whether column is a primary key
- **Property** bool \$multipleKey: specifies whether column is a multiple key
- **Property** bool \$foreignKey: specifies whether column is a foreign key
- **Property** bool \$autoIncrement: specifies whether column is auto incrementing
- **Property** bool \$unique: specifies whether column is unique
- **Property** bool \$numeric: specifies whether column is numeric
- **Property** bool \$string: specifies whether column is a string
- **Property** bool \$integer: specifies whether column is an integer
- **Property** bool \$real: specifies whether column is a double
- **Property** bool \$boolean: specifies whether column is true/false
- **Property** bool \$year: specifies whether column is year string
- **Property** bool \$date: specifies whether column is date string
- **Property** bool \$time: specifies whether column is time string



- **Property** bool \$datetime: specifies whether column is a date/time string
- **Property** bool \$blob: specifies whether column is a blob
- **Property** bool \$binary: specifies whether column is binary
- **Author** Darnell Shinbine

Constructor *void* function ColumnSchema::\_\_construct(\$properties) [[line 175](#)]

**Function Parameters:**

- *array* **\$properties** initial properties

## Constructor

- **Access** public

*mixed* function ColumnSchema::\_\_get(\$field) [[line 204](#)]

**Function Parameters:**

- *string* **\$field** name of the field

**returns an object property**

- **Access** public

# Class DataAdapter

[line 21]

## Provides access to a generic data source

- **Package** PHPRum
- **Sub-Package** DB
- **Property** bool \$opened: Specifies whether the connection is open
- **Property** bool \$closed: Specifies whether the connection is closed
- **Author** Darnell Shinbine
- **Abstract Element**

### DataAdapter::\$args

*array* = array() [line 27]

## A collection of connection variables

- **Access** protected

### DataAdapter::\$cacheEnabled

*bool* = false [line 45]

## Specifies whether caching is enabled

- **Access** protected

### DataAdapter::\$cacheExpires

*int* = 300 [line 51]

## Specifies the caching expiration in seconds

- **Access** protected

#### **DataAdapter::\$charset**

*string* = 'utf-8' [[line 39](#)]

**Specifies the character set**

- **Access** protected

#### **DataAdapter::\$collation**

*string* = 'utf8\_general\_ci' [[line 33](#)]

**Specifies the db collation**

- **Access** protected

Constructor *void* function DataAdapter::\_\_construct(\$args) [[line 102](#)]

#### **Function Parameters:**

- *array* **\$args** connection arguments

#### **Constructor**

- **Access** protected

*DatabaseSchema* function *DataAdapter::addTableSchema(&\$tableSchema)* [*line* [426](#)]

**Function Parameters:**

- [TableSchema](#) &\$tableSchema

**creates a TableSchema object**

- **Abstract Element**
- **Access** public

*DatabaseSchema* function *DataAdapter::alterTableSchema(&\$tableSchema)* [*line* [434](#)]

**Function Parameters:**

- [TableSchema](#) &\$tableSchema

**alters a TableSchema object**

- **Abstract Element**
- **Access** public

*TransactionBase* function *DataAdapter::beginTransaction()* [*line* [518](#)]

**creates a Transaction object**

- **Abstract Element**
- **Access** public

*DatabaseSchema* function *DataAdapter::buildSchema()* [[line 418](#)]  
**builds a DataBaseSchema object**

- **Abstract Element**
- **Access** public

*bool* function *DataAdapter::close()* [[line 467](#)]  
**closes an open connection**

- **Abstract Element**
- **Access** public

*DataAdapter* function *DataAdapter::create(\$dsn)* [[line 171](#)]  
**Function Parameters:**

- *string* **\$dsn** connection string

**creates a connection object to a datasource**

Examples:

```
1 adapter=mysql;uid=root;pwd=;server=localhost;database=northwind;charset=utf8;pconnect=false;cache_enabled=  
false;cache_expires=300;  
2 adapter=mssql;uid=root;pwd=;server=localhost;database=northwind;charset=utf8;  
3 adapter=text;format=TabDelimited;source=/northwind.csv;charset=utf8;  
4 adapter=dir;source=/northwind;
```

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

*void function* DataAdapter::delete(&\$ds) [*line* [502](#)]

**Function Parameters:**

- [DataSet](#) &\$ds reference to a DataSet

**attempt to delete a record in the datasource**

- **Abstract Element**
- **Access** public

*void function* DataAdapter::disableCaching(\$expires) [*line* [296](#)]

**Function Parameters:**

- *int* \$expires cache expiration in seconds

**disable caching**

- **Access** public

*DatabaseSchema function* DataAdapter::dropTableSchema(&\$tableSchema) [*line* [442](#)]

**Function Parameters:**

- [TableSchema](#) &\$tableSchema

**drops a TableSchema object**

- **Abstract Element**
- **Access** public

*void* function DataAdapter::enableCaching([\$expires = null]) [*line* [279](#)]

***Function Parameters:***

- *int* **\$expires** cache expiration in seconds

## enable caching

- **Access** public

*string* function DataAdapter::escapeString(\$unescape\_string) [*line* [543](#)]

***Function Parameters:***

- *string* **\$unescape\_string** String to escape

## Returns an escaped string

- **Abstract Element**
- **Access** public

*void* function DataAdapter::execute(\$query) [*line* [308](#)]

***Function Parameters:***

- *string* **\$query** query to execute

## Executes a query procedure on the current connection

- **Access** public

*void* function DataAdapter::executeBatch(\$batch) [*line* [320](#)]

### **Function Parameters:**

- *string* **\$batch** sql batch query

## Executes a query batch or procedure on the current connection

- **Access** public

*void* function DataAdapter::fill(&\$ds, \$ds) [*line* [451](#)]

### **Function Parameters:**

- [DataSet](#) **\$ds** reference to a DataSet object
- [DataSet](#) **&\$ds**

## fetches DataSet from datasource string using source string

- **Abstract Element**
- **Access** public

*int* function DataAdapter::getAffectedRows() [*line* [534](#)]



## return affected rows

- **Abstract Element**
- **Access** public

*int* function DataAdapter::getLastInsertId() [[line 526](#)]  
**return id of last record inserted**

- **Abstract Element**
- **Access** public

*string* function DataAdapter::getLastQuery() [[line 340](#)]  
**return last executed query**

- **Access** public

*real* function DataAdapter::getLastQueryTime() [[line 351](#)]  
**return last executed query time in milliseconds**

- **Access** public

*int* function DataAdapter::getQueryCount() [[line 362](#)]  
**returns the query count**

- **Access** public

*int* function DataAdapter::getQueryTime() [*line* [373](#)]  
**returns the total query time**

- **Access** public

*DatabaseSchema* function DataAdapter::getSchema() [*line* [384](#)]  
**returns a DataBaseSchema object**

- **Access** public

*void* function DataAdapter::insert(&\$ds, \$ds) [*line* [484](#)]  
**Function Parameters:**

- [DataSet](#) **\$ds** reference to a DataSet
- [DataSet](#) **&\$ds**

**attempt to insert a record into the datasource**

- **Abstract Element**
- **Access** public

*bool* function DataAdapter::open() [*line* [459](#)]  
**opens a connection to a datasource**

- **Abstract Element**
- **Access** public

*DataSet* function `DataAdapter::openDataSet([$source = ''], [$lock_type = null])` [*line* [251](#)]

**Function Parameters:**

- *string* **\$source** source string (or QueryBuilder)
- [DataSetType](#) **\$lock\_type** lock type as constant of `DataSetType::OpenDynamic()`, `DataSetType::OpenStatic()`, or `DataSetType::OpenReadOnly()`

**opens a DataSet specified by the source**

- **Access** public

*bool* function `DataAdapter::opened()` [*line* [475](#)]

**returns true if a connection to a datasource is currently open**

- **Abstract Element**
- **Access** public

*resource* function `DataAdapter::query($query, $buffer)` [*line* [553](#)]

**Function Parameters:**

- *string* **\$query** query string
- *bool* **\$buffer** buffer resultset

**Executes a query procedure on the current connection and return the result**

- **Abstract Element**
- **Access** protected

*QueryBuilderBase* function *DataAdapter::queryBuilder()* [[line 510](#)]  
**creates a QueryBuilder object**

- **Abstract Element**
- **Access** public

*void* function *DataAdapter::rebuildSchema()* [[line 407](#)]  
**rebuilds the DataBaseSchema object**

- **Access** public

*resource* function *DataAdapter::runQuery(\$query, [\$buffer = true])* [[line 563](#)]  
**Function Parameters:**

- *string* **\$query** query to execute
- *bool* **\$buffer** buffer resultset

**Executes a query procedure on the current connection and return the result**

- **Access** protected

`void function DataAdapter::update(&$ds)` [[line 493](#)]

**Function Parameters:**

- [DataSet](#) **&\$ds** reference to a DataSet

**attempt to update a record in the datasource**

- **Abstract Element**
- **Access** public

## Class DataAdapterException

[[line 19](#)]

**Represents a DataAdapter Exception**

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Darnell Shinbine

# Class DatabaseException

[[line 19](#)]

**Represents a DataBase Exception**

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Darnell Shinbine

# Class DatabaseSchema

[[line 21](#)]

**Represents a database schema**

- **Package** PHPRum
- **Sub-Package** DB
- **Property** array \$tableSchemas: array of table schemas
- **Property** array \$tableNames: array of table names
- **Author** Darnell Shinbine

Constructor *void* function DatabaseSchema::\_\_construct(\$properties, \$tableSchemas, \$dataAdapter) [[line 36](#)]

**Function Parameters:**

- [DataAdapter](#) **\$dataAdapter** instance of a DataAdapter
- *array* **\$properties**
- *array* **\$tableSchemas**

## Constructor

- **Access** public

*TableSchema* function DatabaseSchema::seek(\$tableName) [[line 95](#)]

### **Function Parameters:**

- *string* **\$tableName** table name

## seek

- **Access** public

*mixed* function DatabaseSchema::\_\_get(\$field) [[line 67](#)]

### **Function Parameters:**

- *string* **\$field** name of the field

## returns an object property

- **Access** public

# Class DataSet

## Represents a DataSet

- **Package** PHPRum
- **Sub-Package** DB
- **Property** int \$page: the current page
- **Property** int \$pageCount: the current page count
- **Property** int \$pageSize: the page size
- **Property** int \$cursor: record pointer
- **Property** int \$count: number of records
- **Property** bool \$eof: specifies if end of record
- **Property** bool \$bof: specifies if beginning of record
- **Property** string \$source: data source string
- **Property** string \$table: source table
- **Property** array \$fieldMeta: array of field meta data
- **Property** array \$fields: array of fields
- **Property** array \$rows: collection of records
- **Property** array \$row: record data
- **Property** [DataAdapter](#) \$dataAdapter: data adapter
- **Author** Darnell Shinbine

*DataSet* function DataSet::addNew(\$source, &\$da, [\$lock\_type = null]) [line [216](#)]

### **Function Parameters:**

- *\$source* **\$source** data source
- *\$da* **&\$da** DataAdapter
- *\$lock\_type* **\$lock\_type** lock type as constant of DataSetType::OpenDynamic(), DataSetType::OpenStatic(), or DataSetType::OpenReadOnly()



## create new DataSet

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

*bool* function DataSet::bof() [[line 813](#)]  
**returns true at beggining of file**

- **Access** public

*void* function DataSet::delete() [[line 1127](#)]  
**Delete record from DataSource**

- **Access** public

*bool* function DataSet::eof() [[line 798](#)]  
**Returns true on end of file**

- **Access** public

*void* function DataSet::filter(\$column, \$operator, [\$value = "], [\$case\_insensitive = false]) [[line 648](#)]  
**Function Parameters:**

- *string* **\$column** column name to filter
- *string* **\$value** field value to search for

- *string* **\$operator** operation to perform
- *int* **\$case\_insensitive** specifies case insensitive

## filter DataSet by removing all records

- **Access** public

*void* function DataSet::first() [[line 497](#)]

## Moves result pointer to first record

- **Access** public

*real* function DataSet::getAvg(\$column) [[line 829](#)]

### Function Parameters:

- *string* **\$column** name of column

## returns average of values in column

- **Access** public

*string* function DataSet::getCSVString([\$enclosure = "\""], [\$fdelimiter = "\t"],  
[\$rdelimiter = "\n"], [\$nullexpr = 'NULL'], [\$fieldnames = true], [\$n = 0]) [[line](#)

[1186](#)]

### Function Parameters:

- *char* **\$enclosure** string enclosure

- *char* **\$fdelimiter** column dilemeter
- *char* **\$rdelimiter** row dilemeter
- *string* **\$nullexpr** null expression
- *bool* **\$fieldnames** include fields names in string
- *int* **\$n** no of rows

**Returns a DataSet as a CSV string**

- **Access** public

*real* function DataSet::getMax(\$column) [[line 868](#)]

**Function Parameters:**

- *string* **\$column** name of column

**returns max of values in column**

- **Access** public

*real* function DataSet::getMin(\$column) [[line 938](#)]

**Function Parameters:**

- *string* **\$column** name of column

**returns min of values in column**

- **Access** public

*int* function DataSet::getSum(\$column) [[line 1008](#)]

**Function Parameters:**

- *string* **\$column** name of column

**returns sum of all values in column**

- **Access** public

*string* function DataSet::getXMLString() [[line 1273](#)]

**Returns a DataSet as an XML string**

- **Since** 3.5.5
- **Access** public

*void* function DataSet::insert() [[line 1075](#)]

**Inserts record into DataSource**

- **Access** public

*void* function DataSet::last() [[line 509](#)]

**Moves result pointer to last record**

- **Access** public

*void* function DataSet::move(\$cursor) [*line* [523](#)]

**Function Parameters:**

- *int* **\$cursor** new location of result pointer

**Moves record pointer to specified record**

- **Access** public

*void* function DataSet::next() [*line* [473](#)]

**Moves result pointer to next record**

- **Access** public

*int* function DataSet::page([\$page = null]) [*line* [583](#)]

**Function Parameters:**

- *int* **\$page** page

**Moves record pointer to first record on specified page**

- **Access** public

*int* function DataSet::pageCount() [[line 620](#)]

**Returns the number of pages with data in a DataSet object**

- **Access public**

*void* function DataSet::prev() [[line 485](#)]

**Moves result pointer to previous record**

- **Access public**

*void* function DataSet::requery() [[line 1051](#)]

**Requery datasource**

- **Access public**

*bool* function DataSet::seek(\$field, \$value, [\$case\_insensitive = 0]) [[line 547](#)]

**Function Parameters:**

- *string* **\$field** field name to search on
- *string* **\$value** field value to search for
- *bool* **\$case\_insensitive** specifies case insensitive

**sets result pointer to record based on search criteria**

- **Access** public

*void function DataSet::setFieldMeta(\$fieldMeta, \$fields) [line [327](#)]*

***Function Parameters:***

- *array* **\$fields** fields to add to DataSet
- *array* **\$fieldMeta**

## set fields

- **Access** public

*void function DataSet::setFields(\$fields) [line [308](#)]*

***Function Parameters:***

- *array* **\$fields** fields to add to DataSet

## set fields

- **Access** public

*void function DataSet::setRows(\$rows, \$records) [line [346](#)]*

***Function Parameters:***

- *array* **\$records** records to add to DataSet
- *array* **\$rows**

## set records

- **Access** public

*void* function DataSet::setTable(\$table) [*line* [365](#)]

### **Function Parameters:**

- *string* **\$table** source table

## set table

- **Access** public

*void* function DataSet::sort(\$column, [\$reverse = false], [\$case\_insensitive = false]) [*line* [412](#)]

### **Function Parameters:**

- *string* **\$column** column name
- *bool* **\$reverse** specifies reverse sort order
- *bool* **\$case\_insensitive** specifies if case sensitive

## sorts DataSet

- **Access** public

*array* function DataSet::toArray() [*line* [1169](#)]

### **Returns an array containg all rows**



- **Access** public

*void* function DataSet::update() [*line* [1101](#)]

### **Update DataSource**

- **Access** public

*void* function DataSet::updateRowData() [*line* [383](#)]

### **update accessor with data from current record**

- **Access** public

*mixed* function DataSet::\_\_get(\$field) [*line* [103](#)]

### **Function Parameters:**

- *string* **\$field** name of the field

**returns an object property**

- **Access** public

# Class DataSetType

[[line 18](#)]

**Represents a DataSet type**

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Darnell Shinbine

*DataSetType* function DataSetType::OpenDynamic() [[line 31](#)]

**Specifies the DataSet type is OpenDynamic**

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

*DataSetType* function DataSetType::OpenReadOnly() [[line 43](#)]

**Specifies the DataSet type is OpenReadOnly**

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

*DataSetType* function DataSetType::OpenStatic() [[line 37](#)]

**Specifies the DataSet type is OpenStatic**

- **Static**

- **Access** public

## Class DirDataAdapter

*[line 19]*

**Represents an open connection to the file system**

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Darnell Shinbine

*DatabaseSchema* function DirDataAdapter::addTableSchema(\\&\$tableSchema) *[line 73]*

**Function Parameters:**

- [TableSchema](#) \\&\$tableSchema

**creates a TableSchema object**

- **Access** public

*DatabaseSchema* function DirDataAdapter::alterTableSchema(\\&\$tableSchema) *[line 84]*

**Function Parameters:**

- [TableSchema](#) \\&\$tableSchema

## alters a TableSchema object

- **Access** public

*Transaction* function DirDataAdapter::beginTransaction() [[line 296](#)]  
**creates a Transaction object**

- **Access** public

*DatabaseSchema* function DirDataAdapter::buildSchema() [[line 46](#)]  
**builds a DataBaseSchema object**

- **Access** public

*bool* function DirDataAdapter::close() [[line 220](#)]  
**closes an open connection**

- **Access** public

*void* function DirDataAdapter::delete(\\&\$ds, &\$ds) [[line 274](#)]  
**Function Parameters:**

- [DataSet](#) &\$ds reference to a DataSet
- [DataSet](#) \\&\$ds

## attempt to delete a record in the datasource

- **Access** public

*DatabaseSchema* function DirDataAdapter::dropTableSchema(\\&\$tableSchema) [*line* [95](#)]

### **Function Parameters:**

- [TableSchema](#) \\&\$tableSchema

## drops a TableSchema object

- **Access** public

*string* function DirDataAdapter::escapeString(\$unescape\_string) [*line* [330](#)]

### **Function Parameters:**

- *string* **\$unescape\_string** String to escape

## Returns escaped string

- **Access** public

*void* function DirDataAdapter::fill(\\&\$ds, &\$ds) [*line* [107](#)]

### **Function Parameters:**

- [DataSet](#) **&\$ds** empty DataSet object

- [DataSet](#) \\&\$ds

fetches DataSet from datasource string using source string

- Access public

*int* function DirDataAdapter::getAffectedRows() [*line* [318](#)]  
**return affected rows**

- Access public

*int* function DirDataAdapter::getLastInsertId() [*line* [307](#)]  
**return id of last record inserted**

- Access public

*void* function DirDataAdapter::insert(\\&\$ds, \$ds) [*line* [250](#)]  
**Function Parameters:**

- [DataSet](#) \$ds reference to a DataSet
- [DataSet](#) \\&\$ds

**attempt to insert a record into the datasource**

- **Access** public

*bool* function DirDataAdapter::open() [*line* [187](#)]

**opens a connection to a filesystem resource**

- **Access** public

*bool* function DirDataAdapter::opened() [*line* [238](#)]

**returns true if a connection to a datasource is currently open**

- **Access** public

*resource* function DirDataAdapter::query(\$query, \$buffer) [*line* [35](#)]

**Function Parameters:**

- *string* **\$query** sql query
- *bool* **\$buffer** buffer resultset

**Executes a query procedure on the current connection and return the result**

- **Access** protected

*QueryBuilder* function DirDataAdapter::queryBuilder() [*line* [285](#)]

**creates a QueryBuilder object**

- **Access** public

*void* function DirDataAdapter::update(\\&\$ds, &\$ds) [*line* [262](#)]

**Function Parameters:**

- [DataSet](#) &\$ds reference to a DataSet
- [DataSet](#) \\&\$ds

**attempt to update a record in the datasource**

- **Access** public

## Class MSSQLDataAdapter

[*line* [20](#)]

**Represents an open connection to a MSSQL database**

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Tahsin Zulkarnine

**MSSQLDataAdapter::\$charset**

*string* = 'utf8' [*line* [32](#)]

**Specifies the character set**



- **Access** protected

*DatabaseSchema* function MSSQLDataAdapter::addTableSchema(\\&\$tableSchema) [*line 74*]

**Function Parameters:**

- [\*TableSchema\*](#) \\&\$tableSchema

**creates a TableSchema object**

- **Access** public

*DatabaseSchema* function MSSQLDataAdapter::alterTableSchema(\\&\$tableSchema) [*line 186*]

**Function Parameters:**

- [\*TableSchema\*](#) \\&\$tableSchema

**alters a TableSchema object**

- **Access** public

*void* function MSSQLDataAdapter::begin(\$query) [*line 719*]

**Function Parameters:**

- *string* **\$query** sql query

**Executes a sql query or procedure on the current connection**

- **Access** public

*MSSQLTransaction* function MSSQLDataAdapter::beginTransaction() [[line 337](#)]  
**creates a Transaction object**

- **Access** public

*DatabaseSchema* function MSSQLDataAdapter::buildSchema() [[line 213](#)]  
**builds a DataBaseSchema object**

- **Access** public

*bool* function MSSQLDataAdapter::close() [[line 570](#)]  
**closes an open connection**

- **Access** public

*void* function MSSQLDataAdapter::commit() [[line 755](#)]  
**commit transaction**

- **Access** public

*void* function MSSQLDataAdapter::delete(\\&\$ds, &\$ds) [*line* [676](#)]

**Function Parameters:**

- [DataSet](#) &\$ds reference to a DataSet
- [DataSet](#) \\&\$ds

**attempt to delete a record in the datasource**

- **Access** public

*DatabaseSchema* function MSSQLDataAdapter::dropTableSchema(\\&\$tableSchema) [*line* [195](#)]

**Function Parameters:**

- [TableSchema](#) \\&\$tableSchema

**drops a TableSchema object**

- **Access** public

*string* function MSSQLDataAdapter::escapeString(\$unescaped\_string) [*line* [382](#)]

**Function Parameters:**

- *string* \$unescaped\_string String to escape

**Returns escaped string**

- **Access** public

*resource* function MSSQLDataAdapter::executeInternal(\$query) [[line 861](#)]

**Function Parameters:**

- *string* **\$query** sql query

**Executes a sql query or procedure on the current connection**

- **Access** protected

*void* function MSSQLDataAdapter::fill(\\&\$ds, &\$ds) [[line 395](#)]

**Function Parameters:**

- [DataSet](#) **&\$ds** empty DataSet object
- [DataSet](#) **\\&\$ds**

**fetches DataSet from database string using source string**

- **Access** public

*int* function MSSQLDataAdapter::getAffectedRows() [[line 842](#)]

**return affected rows**

- **Access** public

*void* function MSSQLDataAdapter::getField(\$table, \$fieldName) [*line* [1202](#)]

**Function Parameters:**

- **\$table**
- **\$fieldName**

- **Access** public

*int* function MSSQLDataAdapter::getLastColumnIdOfTable(\$table, \$column) [*line* [822](#)]

**Function Parameters:**

- **\$table**
- **\$column**

**return id of last record inserted for a column in table**

- **Access** public

*int* function MSSQLDataAdapter::getLastInsertId() [*line* [803](#)]

**return id of last record inserted**

- **Access** public

*string* function MSSQLDataAdapter::getQuery(\$queryBuilder) [*line* [907](#)]

**Function Parameters:**

- *QueryBuilder* **\$queryBuilder** QueryBuilder object

**Returns SQL based on the QueryBuilder**

- **Access** protected

*DataSet* function MSSQLDataAdapter::getTables() [*line* [792](#)]  
**returns a DataSet of tables**

- **Access** public

*void* function MSSQLDataAdapter::insert(\\&\$ds, &\$ds) [*line* [597](#)]  
**Function Parameters:**

- [DataSet](#) **&\$ds** reference to a DataSet
- [DataSet](#) **\\&\$ds**

**attempt to insert a record into the datasource**

- **Access** public

*bool* function MSSQLDataAdapter::open() [*line* [38](#)]  
**opens a connection to a mysql database**

- **Access** public

*bool* function MSSQLDataAdapter::opened() [[line 204](#)]

**returns true if a connection to a datasource is currently open**

- **Access** public

*resource* function MSSQLDataAdapter::query(\$query, \$buffer) [[line 357](#)]

**Function Parameters:**

- *string* **\$query** sql query
- *bool* **\$buffer** buffer resultset

**Executes a query procedure on the current connection and return the result**

- **Access** protected

*MySQLQueryBuilder* function MSSQLDataAdapter::queryBuilder() [[line 346](#)]

**creates a QueryBuilder object**

- **Access** public

*void* function MSSQLDataAdapter::rollback() [[line 737](#)]

**rollback transaction**

- **Access** public

*void function MSSQLDataAdapter::setForeignKeyChecks(\$set) [line [774](#)]*

***Function Parameters:***

- *bool* **\$set** set/unset foreign key checks

## set foreign key checks

- **Access** public

*void function MSSQLDataAdapter::update(\\&\$ds, &\$ds) [line [644](#)]*

***Function Parameters:***

- [DataSet](#) **&\$ds** reference to a DataSet
- [DataSet](#) **\\&\$ds**

## attempt to update a record in the datasource

- **Access** public

# Class MSSQLQueryBuilder

[line [21](#)]



## Represents an SQL Query

- **Package** PHPRum
- **Sub-Package** DB
- **Property** bool \$empty: specifies whether to return empty result set
- **Author** Darnell Shinbine

*void function MSSQLQueryBuilder::addColumn([\$table = '\*'], [\$column = '\*'], [\$alias = ''])* [*line [77](#)*]

### ***Function Parameters:***

- **\$table**
- **\$column**
- **\$alias**

## add column

- **Access** protected

*void function MSSQLQueryBuilder::addGroupByClause(\$table, \$column)* [*line [169](#)*]

### ***Function Parameters:***

- **\$table**
- **\$column**

## add group by clause

- **Access** protected

*void* function MSSQLQueryBuilder::addHavingClause(\$column, \$operand, \$value) [[line 156](#)]

**Function Parameters:**

- **\$column**
- **\$operand**
- **\$value**

## add having clause

- **Access** protected

*void* function MSSQLQueryBuilder::addJoin(\$type, \$lefttable, \$leftcolumn, \$righttable, \$rightcolumn, [\$alias = ""]) [[line 113](#)]

**Function Parameters:**

- **\$type**
- **\$lefttable**
- **\$leftcolumn**
- **\$righttable**
- **\$rightcolumn**
- **\$alias**

## add join

- **Access** protected

*void function MSSQLQueryBuilder::addOrderByClause(\$table, \$column, [\$direction = 'asc'])* [*line [143](#)*]

***Function Parameters:***

- **\$table**
- **\$column**
- **\$direction**

## **add order by clause**

- **Access** protected

*void function MSSQLQueryBuilder::addTable(\$table, [\$alias = ''])* [*line [101](#)*]

***Function Parameters:***

- **\$table**
- **\$alias**

## **add table**

- **Access** protected

*void function MSSQLQueryBuilder::addValue(\$value)* [*line [90](#)*]

***Function Parameters:***

- **\$value**

## add value

- **Access** protected

*void* function MSSQLQueryBuilder::addWhereClause(\$table, \$column, \$operand, \$value) [*line* [129](#)]

### **Function Parameters:**

- **\$table**
- **\$column**
- **\$operand**
- **\$value**

## add where clause

- **Access** protected

*string* function MSSQLQueryBuilder::getQuery() [*line* [181](#)]

### **get SQL query**

- **Access** public

*void* function MSSQLQueryBuilder::is\_date(\$str) [*line* [493](#)]

### **Function Parameters:**

- **\$str**

# Class MSSQLTransaction

[line [20](#)]

**Represents an open connection to a MSSQL database**

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Tahsin Zulkarnine

*void* function MSSQLTransaction::beginTransaction() [line [25](#)]

**Begins a transaction**

- **Access** protected

*void* function MSSQLTransaction::commitTransaction() [line [44](#)]

**Implements a commit**

- **Access** protected

*void* function MSSQLTransaction::rollbackTransaction() [line [35](#)]

**Implements a rollback**

- **Access** protected

## Class MySQLDataAdapter

*[line [19](#)]*

**Represents an open connection to a MySQL database**

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Darnell Shinbine

**MySQLDataAdapter::\$charset**

*string = 'utf8' [line [31](#)]*

**Specifies the character set**

- **Access** protected

*DatabaseSchema* function MySQLDataAdapter::addTableSchema(\\&\$tableSchema) *[line [296](#)]*

**Function Parameters:**

- [TableSchema](#) \\&\$tableSchema

**creates a TableSchema object**

- **Access** public

*DatabaseSchema* function MySQLDataAdapter::alterTableSchema(\\&\$tableSchema) [*line* [407](#)]

**Function Parameters:**

- [TableSchema](#) \\&\$tableSchema

**alters a TableSchema object**

- **Access** public

*MySQLTransaction* function MySQLDataAdapter::beginTransaction() [*line* [532](#)]

**creates a Transaction object**

- **Access** public

*DatabaseSchema* function MySQLDataAdapter::buildSchema() [*line* [235](#)]

**builds a DataBaseSchema object**

- **Access** public

*bool* function MySQLDataAdapter::close() [*line* [101](#)]

**closes an open connection**

- **Access** public

*void* function MySQLDataAdapter::delete(\\&\$ds, &\$ds) [*line* [490](#)]

**Function Parameters:**

- [DataSet](#) &\$ds empty DataSet object
- [DataSet](#) \\&\$ds

**attempt to delete a record in the datasource**

- **Access** public

*DatabaseSchema* function MySQLDataAdapter::dropTableSchema(\\&\$tableSchema) [*line* [418](#)]

**Function Parameters:**

- [TableSchema](#) \\&\$tableSchema

**drops a TableSchema object**

- **Access** public

*string* function MySQLDataAdapter::escapeString(\$unescaped\_string) [*line* [580](#)]

**Function Parameters:**

- *string* \$unescaped\_string String to escape

**Returns escaped string**



- **Access** public

*void* function MySQLDataAdapter::fill(\\&\$ds, &\$ds) [*line* [166](#)]

**Function Parameters:**

- [DataSet](#) &\$ds empty DataSet object
- [DataSet](#) \\&\$ds

**fills a DataSet object with the current DataAdapter**

- **Access** public

*int* function MySQLDataAdapter::getAffectedRows() [*line* [561](#)]

**return affected rows**

- **Access** public

*int* function MySQLDataAdapter::getLastInsertId() [*line* [543](#)]

**return id of last record inserted**

- **Access** public

*void* function MySQLDataAdapter::insert(\\&\$ds, &\$ds) [*line* [430](#)]

**Function Parameters:**

- [DataSet](#) &\$ds empty DataSet object
- [DataSet](#) \\&\$ds

**attempt to insert a record into the datasource**

- **Access** public

*bool* function MySQLDataAdapter::open() [*line 38*]  
**opens a connection to a mysql database**

- **Access** public

*bool* function MySQLDataAdapter::opened() [*line 127*]  
**returns true if a connection to a datasource is currently open**

- **Access** public

*resource* function MySQLDataAdapter::query(\$query, \$buffer) [*line 140*]  
**Function Parameters:**

- *string* **\$query** sql query
- *bool* **\$buffer** buffer resultset

**Executes a query procedure on the current connection and return the result**

- **Access** protected

*MySQLQueryBuilder* function MySQLDataAdapter::queryBuilder() [*line 521*]  
**creates a QueryBuilder object**

- **Access** public

*void* function MySQLDataAdapter::update(\\&\$ds, &\$ds) [*line 458*]  
**Function Parameters:**

- [\*DataSet\*](#) &\$ds empty DataSet object
- [\*DataSet\*](#) \\&\$ds

**attempt to update a record in the datasource**

- **Access** public

## Class MySQLiDataAdapter

[*line 19*]

**Represents an open connection to a MySQL database using the mysqli driver**

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Darnell Shinbine

## MySQLiDataAdapter::\$charset

*string* = 'utf8' [*line* [31](#)]

**Specifies the character set**

- **Access** protected

*DatabaseSchema* function MySQLiDataAdapter::addTableSchema(\\&\$tableSchema) [*line* [243](#)]

**Function Parameters:**

- [TableSchema](#) \\&\$tableSchema

**creates a TableSchema object**

- **Access** public

*DatabaseSchema* function MySQLiDataAdapter::alterTableSchema(\\&\$tableSchema) [*line* [354](#)]

**Function Parameters:**

- [TableSchema](#) \\&\$tableSchema

**alters a TableSchema object**

- **Access** public

*MySQLTransaction* function MySQLiDataAdapter::beginTransaction() [*line* [479](#)]  
**creates a Transaction object**

- **Access** public

*DatabaseSchema* function MySQLiDataAdapter::buildSchema() [*line* [199](#)]  
**builds a DataBaseSchema object**

- **Access** public

*bool* function MySQLiDataAdapter::close() [*line* [84](#)]  
**closes an open connection**

- **Access** public

*void* function MySQLiDataAdapter::delete(\\&\$ds, &\$ds) [*line* [437](#)]  
**Function Parameters:**

- [DataSet](#) &\$ds empty DataSet object
- [DataSet](#) \\&\$ds

**attempt to delete a record in the datasource**

- **Access** public

*DatabaseSchema* function MySQLiDataAdapter::dropTableSchema(\\&\$tableSchema) [*line* [365](#)]

**Function Parameters:**

- [TableSchema](#) \\&\$tableSchema

**drops a TableSchema object**

- **Access** public

*string* function MySQLiDataAdapter::escapeString(\$unescaped\_string) [*line* [527](#)]

**Function Parameters:**

- *string* **\$unescaped\_string** String to escape

**Returns escaped string**

- **Access** public

*void* function MySQLiDataAdapter::fill(\\&\$ds, &\$ds) [*line* [149](#)]

**Function Parameters:**

- [DataSet](#) **&\$ds** empty DataSet object
- [DataSet](#) \\&\$ds

**fills a DataSet object with the current DataAdapter**

- **Access** public

*int* function MySQLiDataAdapter::getAffectedRows() [*line* [508](#)]  
**return affected rows**

- **Access** public

*int* function MySQLiDataAdapter::getLastInsertId() [*line* [490](#)]  
**return id of last record inserted**

- **Access** public

*void* function MySQLiDataAdapter::insert(\\&\$ds, &\$ds) [*line* [377](#)]  
**Function Parameters:**

- [DataSet](#) &\$ds empty DataSet object
- [DataSet](#) \\&\$ds

**attempt to insert a record into the datasource**

- **Access** public

*bool* function MySQLiDataAdapter::open() [*line* [38](#)]

## opens a connection to a mysql database

- **Access** public

*bool* function MySQLiDataAdapter::opened() [*line* [110](#)]

**returns true if a connection to a datasource is currently open**

- **Access** public

*resource* function MySQLiDataAdapter::query(\$query, \$buffer) [*line* [123](#)]

### **Function Parameters:**

- *string* **\$query** sql query
- *bool* **\$buffer** buffer resultset

**Executes a query procedure on the current connection and return the result**

- **Access** protected

*MySQLQueryBuilder* function MySQLiDataAdapter::queryBuilder() [*line* [468](#)]

**creates a QueryBuilder object**

- **Access** public



*void function MySQLiDataAdapter::update(\\&\$ds, &\$ds) [line [405](#)]*

***Function Parameters:***

- [DataSet](#) &\$ds empty DataSet object
- [DataSet](#) \\&\$ds

**attempt to update a record in the datasource**

- **Access** public

## Class MySQLiQueryBuilder

*[line [21](#)]*

**Represents an SQL Query**

- **Package** PHPRum
- **Sub-Package** DB
- **Property** bool \$empty: specifies whether to return empty result set
- **Author** Darnell Shinbine

*void function MySQLiQueryBuilder::addColumn([\$table = '\*'], [\$column = '\*'], [\$alias = '']) [line [77](#)]*

***Function Parameters:***

- **\$table**
- **\$column**
- **\$alias**

## add column

- **Access** protected

*void* function MySQLiQueryBuilder::addGroupByClause(\$table, \$column) [*line* [168](#)]

### ***Function Parameters:***

- **\$table**
- **\$column**

## add group by clause

- **Access** protected

*void* function MySQLiQueryBuilder::addHavingClause(\$column, \$operand, \$value) [*line* [155](#)]

### ***Function Parameters:***

- **\$column**
- **\$operand**
- **\$value**

## add having clause

- **Access** protected

*void* function MySQLiQueryBuilder::addJoin(\$type, \$lefttable, \$leftcolumn, \$righttable, \$rightcolumn, [\$alias = ''])  
[*line* [112](#)]

**Function Parameters:**

- **\$type**
- **\$lefttable**
- **\$leftcolumn**
- **\$righttable**
- **\$rightcolumn**
- **\$alias**

**add join**

- **Access** protected

*void* function MySQLiQueryBuilder::addOrderByClause(\$table, \$column, [\$direction = 'asc']) [*line* [142](#)]

**Function Parameters:**

- **\$table**
- **\$column**
- **\$direction**

**add order by clause**

- **Access** protected

*void function MySQLiQueryBuilder::addTable(\$table, [\$alias = "]) [line [100](#)]*

***Function Parameters:***

- **\$table**
- **\$alias**

## **add table**

- **Access protected**

*void function MySQLiQueryBuilder::addValue(\$value) [line [90](#)]*

***Function Parameters:***

- **\$value**

## **add value**

- **Access protected**

*void function MySQLiQueryBuilder::addWhereClause(\$table, \$column, \$operand, \$value) [line [128](#)]*

***Function Parameters:***

- **\$table**
- **\$column**
- **\$operand**
- **\$value**

## add where clause

- **Access** protected

*string* function MySQLiQueryBuilder::getQuery() [*line* [180](#)]

## get SQL query

- **Access** public

# Class MySQLiTransaction

[*line* [19](#)]

**Represents an open connection to a MySQL database**

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Darnell Shinbine

*void* function MySQLiTransaction::beginTransaction() [*line* [24](#)]

## Begins a transaction

- **Access** protected

*void function MySQLiTransaction::commitTransaction()* [[line 43](#)]

### **Implements a commit**

- **Access** protected

*void function MySQLiTransaction::rollbackTransaction()* [[line 34](#)]

### **Implements a rollback**

- **Access** protected

## Class MySQLQueryBuilder

[[line 21](#)]

### **Represents an SQL Query**

- **Package** PHPRum
- **Sub-Package** DB
- **Property** bool \$empty: specifies whether to return empty result set
- **Author** Darnell Shinbine

*void function MySQLQueryBuilder::addColumn([\$table = '\*'], [\$column = '\*'], [\$alias = ''])* [[line 77](#)]

#### **Function Parameters:**

- **\$table**

- **\$column**
- **\$alias**

## **add column**

- **Access** protected

*void* function MySQLQueryBuilder::addGroupByClause(\$table, \$column) [*line* [168](#)]

### ***Function Parameters:***

- **\$table**
- **\$column**

## **add group by clause**

- **Access** protected

*void* function MySQLQueryBuilder::addHavingClause(\$column, \$operand, \$value) [*line* [155](#)]

### ***Function Parameters:***

- **\$column**
- **\$operand**
- **\$value**

## **add having clause**

- **Access** protected

*void* function MySQLQueryBuilder::addJoin(\$type, \$lefttable, \$leftcolumn, \$righttable, \$rightcolumn, [\$alias = '']) [[line 112](#)]

***Function Parameters:***

- **\$type**
- **\$lefttable**
- **\$leftcolumn**
- **\$righttable**
- **\$rightcolumn**
- **\$alias**

## **add join**

- **Access** protected

*void* function MySQLQueryBuilder::addOrderByClause(\$table, \$column, [\$direction = 'asc']) [[line 142](#)]

***Function Parameters:***

- **\$table**
- **\$column**
- **\$direction**

## **add order by clause**



- **Access** protected

*void* function MySQLQueryBuilder::addTable(\$table, [\$alias = ""]) [*line* [100](#)]

***Function Parameters:***

- **\$table**
- **\$alias**

## **add table**

- **Access** protected

*void* function MySQLQueryBuilder::addValue(\$value) [*line* [90](#)]

***Function Parameters:***

- **\$value**

## **add value**

- **Access** protected

*void* function MySQLQueryBuilder::addWhereClause(\$table, \$column, \$operand, \$value) [*line* [128](#)]

***Function Parameters:***

- **\$table**
- **\$column**
- **\$operand**

- **\$value**

**add where clause**

- **Access** protected

*string* function MySQLQueryBuilder::getQuery() [*line* [180](#)]  
**get SQL query**

- **Access** public

## Class MySQLTransaction

[*line* [19](#)]

**Represents an open connection to a MySQL database**

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Darnell Shinbine

*void* function MySQLTransaction::beginTransaction() [*line* [24](#)]  
**Begins a transaction**

- **Access** protected

*void function MySQLTransaction::commitTransaction()* [[line 43](#)]

**Implements a commit**

- **Access** protected

*void function MySQLTransaction::rollbackTransaction()* [[line 34](#)]

**Implements a rollback**

- **Access** protected

## Class PDODataAdapter

[[line 19](#)]

**Represents an open connection to a MySQL database using the mysqli driver**

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Darnell Shinbine

*DatabaseSchema function PDODataAdapter::addTableSchema(\\&\$tableSchema)* [[line 225](#)]

**Function Parameters:**

- [TableSchema](#) \\&\$tableSchema

**creates a TableSchema object**

- **Access** public

*DatabaseSchema* function PDODataAdapter::alterTableSchema(\\&\$tableSchema) [*line* [336](#)]

**Function Parameters:**

- [TableSchema](#) \\&\$tableSchema

**alters a TableSchema object**

- **Access** public

*MySQLTransaction* function PDODataAdapter::beginTransaction() [*line* [446](#)]

**creates a Transaction object**

- **Access** public

*DatabaseSchema* function PDODataAdapter::buildSchema() [*line* [178](#)]

**builds a MSSQL DataBaseSchema object**

- **Access** public

*bool* function PDODataAdapter::close() [*line* [69](#)]  
**closes an open connection**

- **Access** public

*void* function PDODataAdapter::delete(\\&\$ds, &\$ds) [*line* [411](#)]  
**Function Parameters:**

- [DataSet](#) &\$ds empty DataSet object
- [DataSet](#) \\&\$ds

**attempt to delete a record in the datasource**

- **Access** public

*DatabaseSchema* function PDODataAdapter::dropTableSchema(\\&\$tableSchema) [*line* [347](#)]  
**Function Parameters:**

- [TableSchema](#) \\&\$tableSchema

**drops a TableSchema object**

- **Access** public

*string* function PDODataAdapter::escapeString(\$unescape\_string) [*line* [501](#)]

**Function Parameters:**

- *string* **\$unescape\_string** String to escape

**Returns escaped string**

- **Access** public

*void* function PDODataAdapter::fill(\\&\$ds, &\$ds) [*line* [127](#)]

**Function Parameters:**

- [DataSet](#) **&\$ds** empty DataSet object
- [DataSet](#) **\\&\$ds**

**fills a DataSet object with the current DataAdapter**

- **Access** public

*int* function PDODataAdapter::getAffectedRows() [*line* [482](#)]

**return affected rows**

- **Access** public

*void* function PDODataAdapter::getField(\$table, \$fieldName) [*line* [674](#)]

**Function Parameters:**

- **\$table**
- **\$fieldName**

- **Access** public

*int* function PDODataAdapter::getLastInsertId() [*line* [464](#)]  
**return id of last record inserted**

- **Access** public

*void* function PDODataAdapter::insert(\\&\$ds, &\$ds) [*line* [359](#)]  
**Function Parameters:**

- [DataSet](#) &\$ds empty DataSet object
- [DataSet](#) \\&\$ds

**attempt to insert a record into the datasource**

- **Access** public

*bool* function PDODataAdapter::open() [*line* [33](#)]  
**opens a connection to a mysql database**

- **Access** public

*bool* function PDODataAdapter::opened() [*line [88](#)*]

**returns true if a connection to a datasource is currently open**

- **Access** public

*resource* function PDODataAdapter::query(\$query, \$buffer) [*line [100](#)*]

**Function Parameters:**

- *string* **\$query** sql query
- **\$buffer**

**Executes a query procedure on the current connection and return the result**

- **Access** protected

*MySQLQueryBuilder* function PDODataAdapter::queryBuilder() [*line [434](#)*]

**creates a QueryBuilder object**

- **Access** public

*void* function PDODataAdapter::update(\\&\$ds, &\$ds) [*line [386](#)*]

**Function Parameters:**

- [\*DataSet\*](#) **&\$ds** empty DataSet object
- [\*DataSet\*](#) **\\&\$ds**



attempt to update a record in the datasource

- **Access** public

## Class PDOQueryBuilder

[[line 21](#)]

Represents an SQL Query

- **Package** PHPRum
- **Sub-Package** DB
- **Property** bool \$empty: specifies whether to return empty result set
- **Author** Tahsin Zulkarnine

*void* function PDOQueryBuilder::addColumn([\$table = '\*'], [\$column = '\*'], [\$alias = '']) [[line 77](#)]

**Function Parameters:**

- **\$table**
- **\$column**
- **\$alias**

**add column**

- **Access** protected

*void* function PDOQueryBuilder::addGroupByClause(\$table, \$column) [*line* [168](#)]

***Function Parameters:***

- **\$table**
- **\$column**

## **add group by clause**

- **Access** protected

*void* function PDOQueryBuilder::addHavingClause(\$column, \$operand, \$value) [*line* [155](#)]

***Function Parameters:***

- **\$column**
- **\$operand**
- **\$value**

## **add having clause**

- **Access** protected

*void* function PDOQueryBuilder::addJoin(\$type, \$lefttable, \$leftcolumn, \$righttable, \$rightcolumn, [\$alias = ""]) [*line* [112](#)]

***Function Parameters:***

- **\$type**
- **\$lefttable**
- **\$leftcolumn**
- **\$righttable**
- **\$rightcolumn**
- **\$alias**

## add join

- **Access** protected

*void* function PDOQueryBuilder::addOrderByClause(\$table, \$column, [\$direction = 'asc']) [*line* [142](#)]

### **Function Parameters:**

- **\$table**
- **\$column**
- **\$direction**

## add order by clause

- **Access** protected

*void* function PDOQueryBuilder::addTable(\$table, [\$alias = '']) [*line* [100](#)]

### **Function Parameters:**

- **\$table**
- **\$alias**

## add table

- **Access** protected

*void* function PDOQueryBuilder::addValue(\$value) [*line* [90](#)]

### ***Function Parameters:***

- **\$value**

## add value

- **Access** protected

*void* function PDOQueryBuilder::addWhereClause(\$table, \$column, \$operand, \$value) [*line* [128](#)]

### ***Function Parameters:***

- **\$table**
- **\$column**
- **\$operand**
- **\$value**

## add where clause

- **Access** protected

*string* function PDOQueryBuilder::getMSSQLQuery() [*line [490](#)*]

### **get MSSQL SQL query**

- **Access** public

*string* function PDOQueryBuilder::getMySQLQuery() [*line [191](#)*]

### **get MySQL SQL query**

- **Access** public

*string* function PDOQueryBuilder::getQuery() [*line [180](#)*]

### **get SQL query**

- **Access** public

*void* function PDOQueryBuilder::is\_date(\$str) [*line [802](#)*]

### **Function Parameters:**

- **\$str**

## Class PDOTransaction

[*line [19](#)*]

**Represents an open connection to a PDO**

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Darnell Shinbine

*void* function PDOTransaction::beginTransaction() [[line 24](#)]

### **Begins a transaction**

- **Access** protected

*void* function PDOTransaction::commitTransaction() [[line 42](#)]

### **Implements a commit**

- **Access** protected

*void* function PDOTransaction::rollbackTransaction() [[line 33](#)]

### **Implements a rollback**

- **Access** protected

# Class QueryBuilderInterface

[line 20]

Represents an database query

- **Package** PHPRum
- **Sub-Package** DB
- **Property** bool \$empty: specifies whether to return empty result set
- **Author** Darnell Shinbine
- **Abstract Element**

**QueryBuilderBase::\$dataAdapter**

*DataAdapter* = null [line 38]

Contains a reference to a DataAdapter object

- **Access** protected

**QueryBuilderBase::\$empty**

*bool* = false [line 32]

specifies whether to return empty resultset

- **Access** protected

**QueryBuilderBase::\$statement**

*string* = "" [line 26]

**statement**

- **Access** protected

Constructor *void* function QueryBuilderInterface::\_\_construct(&\$dataAdapter, \$dataAdapter) [*line* [47](#)]

**Function Parameters:**

- [DataAdapter](#) **\$dataAdapter** instance of a DataAdapter
- [DataAdapter](#) &**\$dataAdapter**

## Constructor

- **Access** public

*void* function QueryBuilderInterface::addColumn([\$table = '\*'], [\$column = '\*'], [\$alias = '']) [*line* [410](#)]

**Function Parameters:**

- **\$table**
- **\$column**
- **\$alias**

## add column

- **Abstract Element**
- **Access** protected



*void function* QueryBuilderInterface::addGroupByClause(\$table, \$column) [*line* [466](#)]

***Function Parameters:***

- **\$table**
- **\$column**

**add group by clause**

- **Abstract Element**
- **Access** protected

*void function* QueryBuilderInterface::addHavingClause(\$column, \$operand, \$value) [*line* [458](#)]

***Function Parameters:***

- **\$column**
- **\$operand**
- **\$value**

**add having clause**

- **Abstract Element**
- **Access** protected

*void function* QueryBuilderInterface::addJoin(\$type, \$lefttable, \$leftcolumn, \$righttable, \$rightcolumn, [\$alias = ""]) [*line* [434](#)]

***Function Parameters:***

- **\$type**

- **\$lefttable**
- **\$leftcolumn**
- **\$righttable**
- **\$rightcolumn**
- **\$alias**

## add join

- **Abstract Element**
- **Access** protected

*void function QueryBuilderBase::addOrderByClause(\$table, \$column, [\$direction = 'asc']) [line [450](#)]*

### **Function Parameters:**

- **\$table**
- **\$column**
- **\$direction**

## add order by clause

- **Abstract Element**
- **Access** protected

*void function QueryBuilderBase::addTable(\$table, [\$alias = '']) [line [426](#)]*

### **Function Parameters:**

- **\$table**

- **\$alias**

## **add table**

- **Abstract Element**
- **Access** protected

*void function* QueryBuilderBase::addValue(\$value) [*line* [418](#)]

### ***Function Parameters:***

- **\$value**

## **add value**

- **Abstract Element**
- **Access** protected

*void function* QueryBuilderBase::addWhereClause(\$table, \$column, \$operand, \$value) [*line* [442](#)]

### ***Function Parameters:***

- **\$table**
- **\$column**
- **\$operand**
- **\$value**

## **add where clause**

- **Abstract Element**
- **Access** protected

*QueryBuilder* function QueryBuilderInterface::delete() [[line 182](#)]  
**impliments `delete` statement**

- **Access** public

*QueryBuilder* function QueryBuilderInterface::from(\$table, [\$alias = ""]) [[line 195](#)]  
**Function Parameters:**

- *string* **\$table** table name
- *string* **\$alias** table alias

**impliments `from` statement**

- **Access** public

*string* function QueryBuilderInterface::getQuery() [[line 116](#)]  
**get query**

- **Abstract Element**
- **Access** public

*QueryBuilder* function *QueryBuilderBase::groupBy*(\$table, \$column) [*line* [357](#)]

**Function Parameters:**

- *string* **\$table** table name
- *string* **\$column** column name

**impliments `group by` statement**

- **Access** public

*QueryBuilder* function *QueryBuilderBase::having*(\$column, \$operand, \$value) [*line* [343](#)]

**Function Parameters:**

- *string* **\$column** column name
- *string* **\$operand** operation to perform
- *string* **\$value** column value

**impliments `having` statement**

- **Access** public

*QueryBuilder* function *QueryBuilderBase::innerJoin*(\$lefttable, \$leftcolumn, \$righttable, \$rightcolumn, [\$alias = ""]) [*line* [263](#)]

**Function Parameters:**

- *string* **\$lefttable** left table name
- *string* **\$leftcolumn** left column name

- *string* **\$righttable** right table name
- *string* **\$rightcolumn** right column name
- *string* **\$alias** left table alias

## implements `inner join` statement

- **Access** public

*QueryBuilder* function *QueryBuilderBase::insertInto(\$table, \$columns)* [[line 141](#)]

### **Function Parameters:**

- *string* **\$table** table name
- *array* **\$columns** array of columns

## implements `insert into` statement

- **Access** public

*QueryBuilder* function *QueryBuilderBase::leftJoin(\$lefttable, \$leftcolumn, \$righttable, \$rightcolumn, [\$alias = ''])* [[line 280](#)]

### **Function Parameters:**

- *string* **\$lefttable** left table name
- *string* **\$leftcolumn** left column name
- *string* **\$righttable** right table name
- *string* **\$rightcolumn** right column name
- *string* **\$alias** left table alias

## implements `left join` statement

- **Access** public

*DataSet* function *QueryBuilderBase::openDataSet*([\$lock\_type = null]) [*line* [105](#)]

### **Function Parameters:**

- [DataSetType](#) **\$lock\_type** lock type as constant of *DataSetType::OpenDynamic()*, *DataSetType::OpenStatic()*, or *DataSetType::OpenReadonly()*

## open a DataSet

- **Access** public

*QueryBuilder* function *QueryBuilderBase::orderBy*(\$table, \$column, [\$direction = 'asc']) [*line* [328](#)]

### **Function Parameters:**

- *string* **\$table** table name
- *string* **\$column** column name
- *string* **\$direction** order by direction

## implements `order by` statement

- **Access** public

*QueryBuilder* function `QueryBuilderBase::rightJoin($lefttable, $leftcolumn, $righttable, $rightcolumn, [$alias = ''])` [[line 297](#)]

**Function Parameters:**

- *string* **\$lefttable** left table name
- *string* **\$leftcolumn** left column name
- *string* **\$righttable** right table name
- *string* **\$rightcolumn** right column name
- *string* **\$alias** left table alias

**impliments `right join` statement**

- **Access** public

*void* function `QueryBuilderBase::runQuery()` [[line 93](#)]

**run query**

- **Access** public

*QueryBuilder* function `QueryBuilderBase::select([$table = '*'], [$column = '*'], [$alias = ''])` [[line 127](#)]

**Function Parameters:**

- *string* **\$table** table name
- *string* **\$column** column name
- *string* **\$alias** column alias

**impliments `select` statement**



- **Access** public

*QueryBuilder* function QueryBuilderInterface::set(\$table, \$column, \$value) [[line 225](#)]

**Function Parameters:**

- *string* **\$table** table name
- *string* **\$column** column name
- *string* **\$value** column value

**impliments `set` statement**

- **Access** public

*QueryBuilder* function QueryBuilderInterface::setColumns(\$table, \$columns, \$values) [[line 241](#)]

**Function Parameters:**

- *string* **\$table** table name
- *array* **\$columns** column names
- *array* **\$values** values

**impliments `set` statement**

- **Access** public

*QueryBuilder* function QueryBuilderInterface::truncate(\$table) [[line 170](#)]

#### ***Function Parameters:***

- *string* **\$table** table name

#### **impliments `truncate` statement**

- **Access** public

*QueryBuilder* function *QueryBuilderBase::update(\$table)* [[line 157](#)]

#### ***Function Parameters:***

- *string* **\$table** table name

#### **impliments `update` statement**

- **Access** public

*QueryBuilder* function *QueryBuilderBase::values(\$values)* [[line 208](#)]

#### ***Function Parameters:***

- *array* **\$values** array of column values

#### **impliments `values` statement**

- **Access** public

*QueryBuilder* function *QueryBuilderBase::where*(\$table, \$column, \$operand, \$value) [[line 313](#)]

**Function Parameters:**

- *string* **\$table** table name
- *string* **\$column** column name
- *string* **\$operand** operation to perform
- *string* **\$value** column value

**impliments `where` statement**

- **Access** public

## Class QueryException

[[line 19](#)]

**Represents a QueryBuilder Exception**

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Darnell Shimbine

# Class SchemaException

[[line 19](#)]

**Represents a Schema Exception**

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Darnell Shinbine

# Class TableSchema

[[line 24](#)]

**Represents a table schema**

- **Package** PHPRum
- **Sub-Package** DB
- **Property** string \$name: table name
- **Property** string \$primaryKey: primary key
- **Property** array \$foreignKeys: array of foreign keys
- **Property** array \$columnSchemas: array of column schemas
- **Property** array \$columnNames: array of column names
- **Author** Darnell Shinbine

Constructor *void* function TableSchema::\_\_construct(\$properties, \$foreignKeys, \$columnSchemas) [[line 59](#)]

**Function Parameters:**

- *array* **\$properties** initial properties

- *array* **\$foreignKeys** array of foreign keys
- *array* **\$columnSchemas** array of column schemas

## Constructor

- **Access** public

*TableSchema* function TableSchema::seek(\$columnName, \$tableName) [*line* [119](#)]

### **Function Parameters:**

- *string* **\$tableName** table name
- **\$columnName**

## seek

- **Access** public

*mixed* function TableSchema::\_\_get(\$field) [*line* [91](#)]

### **Function Parameters:**

- *string* **\$field** name of the field

## returns an object property

- **Access** public

# Class TextDataAdapter

[line [19](#)]

Represents an open connection to a tab or comma delimited text file

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Darnell Shinbine

*DatabaseSchema* function TextDataAdapter::addTableSchema(\\&\$tableSchema) [line [73](#)]

**Function Parameters:**

- [TableSchema](#) \\&\$tableSchema

**creates a TableSchema object**

- **Access** public

*DatabaseSchema* function TextDataAdapter::alterTableSchema(\\&\$tableSchema) [line [84](#)]

**Function Parameters:**

- [TableSchema](#) \\&\$tableSchema

**alters a TableSchema object**

- **Access** public

*Transaction* function TextDataAdapter::beginTransaction() [[line 332](#)]  
**creates a Transaction object**

- **Access** public

*DatabaseSchema* function TextDataAdapter::buildSchema() [[line 46](#)]  
**builds a DataBaseSchema object**

- **Access** public

*bool* function TextDataAdapter::close() [[line 210](#)]  
**closes an open connection**

- **Access** public

*void* function TextDataAdapter::delete(\\&\$ds, &\$ds) [[line 298](#)]  
**Function Parameters:**

- [DataSet](#) &\$ds reference to a DataSet
- [DataSet](#) \\&\$ds

**attempt to delete a record in the datasource**

- **Access** public

*DatabaseSchema* function TextDataAdapter::dropTableSchema(\\&\$tableSchema) [*line 95*]

**Function Parameters:**

- [TableSchema](#) \\&\$tableSchema

**drops a TableSchema object**

- **Access** public

*string* function TextDataAdapter::escapeString(\$unescape\_string) [*line 366*]

**Function Parameters:**

- *string* **\$unescape\_string** String to escape

**Returns escaped string**

- **Access** public

*void* function TextDataAdapter::fill(\\&\$ds, &\$ds) [*line 107*]

**Function Parameters:**

- [DataSet](#) **&\$ds** empty DataSet object
- [DataSet](#) \\&\$ds



fetches DataSet from datasource string using source string

- **Access** public

*int* function TextDataAdapter::getAffectedRows() [*line* [354](#)]  
**return affected rows**

- **Access** public

*int* function TextDataAdapter::getLastInsertId() [*line* [343](#)]  
**return id of last record inserted**

- **Access** public

*void* function TextDataAdapter::insert(\\&\$ds, &\$ds) [*line* [243](#)]  
**Function Parameters:**

- [DataSet](#) &\$ds reference to a DataSet
- [DataSet](#) \\&\$ds

**attempt to insert a record into the datasource**

- **Access** public

*bool* function TextDataAdapter::open() [[line 159](#)]

**opens a connection to a text file**

- **Access** public

*bool* function TextDataAdapter::opened() [[line 231](#)]

**returns true if a connection to a datasource is currently open**

- **Access** public

*resource* function TextDataAdapter::query(\$query, \$buffer) [[line 35](#)]

**Function Parameters:**

- *string* **\$query** sql query
- *bool* **\$buffer** buffer resultset

**Executes a query procedure on the current connection and return the result**

- **Access** protected

*QueryBuilder* function TextDataAdapter::queryBuilder() [[line 321](#)]

**creates a QueryBuilder object**

- **Access** public

*void function TextDataAdapter::update(\\&\$ds, &\$ds) [line [277](#)]*

**Function Parameters:**

- [DataSet](#) **&\$ds** reference to a DataSet
- [DataSet](#) **\\&\$ds**

**attempt to update a record in the datasource**

- **Access** public

## Class TransactionBase

*[line [18](#)]*

**Represents a transaction**

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Darnell Shinbine
- **Abstract Element**

**TransactionBase::\$dataAdapter**

*DataAdapter = null [line [24](#)]*

**Contains a reference to a DataAdapter object**

- **Access** protected

Constructor *void* function TransactionBase::\_\_construct(&\$dataAdapter, \$dataAdapter) [*line* [39](#)]

**Function Parameters:**

- [DataAdapter](#) **\$dataAdapter** instance of a DataAdapter
- [DataAdapter](#) **&\$dataAdapter**

## Constructor

- **Access** public

*void* function TransactionBase::begin() [*line* [49](#)]

### **Begins a transaction**

- **Access** public

*void* function TransactionBase::beginTransaction() [*line* [100](#)]

### **Begins a transaction**

- **Abstract Element**
- **Access** protected

*void* function TransactionBase::commit() [*line* [83](#)]

### **Implements a commit**

- **Access** public

*void* function TransactionBase::commitTransaction() [*line* [112](#)]

**Implements a commit**

- **Abstract Element**
- **Access** protected

*void* function TransactionBase::rollback() [*line* [66](#)]

**Implements a rollback**

- **Access** public

*void* function TransactionBase::rollbackTransaction() [*line* [106](#)]

**Implements a rollback**

- **Abstract Element**
- **Access** protected

# Class TransactionException

[*line* [19](#)]

**Represents a Transaction Exception**

- **Package** PHPRum
- **Sub-Package** DB
- **Author** Darnell Shinbine

# deploy.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# deploymentbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# sshclientbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# get.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class Deploy

[line [19](#)]

### deplomyent script

- **Package** PHPRum
- **Sub-Package** Deploy
- **Author** Darnell Shinbine

*void* function Deploy::execute(\$argc, \$argv) [line [29](#)]

#### **Function Parameters:**

- *int* **\$argc** Number of command line arguments
- *array* **\$argv** Array of command line arguments

### execute the application

- **Access** protected

array function Deploy::getOptions(\$argc, \$argv) [[line 87](#)]

**Function Parameters:**

- *int* **\$argc** Number of command line arguments
- *array* **\$argv** Array of command line arguments

## retrieve command line options

- **Access** protected

void function Deploy::handleError(\$errno, \$errstr, \$errfile, \$errline) [[line 134](#)]

**Function Parameters:**

- *string* **\$errno** error code
- *string* **\$errstr** error description
- *string* **\$errfile** file
- *string* **\$errline** line no.

event triggered by an error in the application, can be overridden to provide error handling.

- **Access** protected

void function Deploy::handleException(\\$e, \$e) [[line 121](#)]

### **Function Parameters:**

- `\Exception $e`
- `Exception $e`

event triggered by an uncaught Exception thrown in the application, can be overridden to provide error handling.

- **Access** protected

## Class DeploymentBase

[line [22](#)]

**Provides base functionality for deploying applications to remote servers.**

The DeploymentBase exposes 2 protected properties, these properties must be defined in the sub class

- **Package** PHPRum
- **Sub-Package** Deploy
- **Property** string \$home\_path: Specifies the home path on the remote server
- **Property** string \$release\_path: Specifies the release path on the remote server
- **Author** Darnell Shinbine
- **Abstract Element**

**DeploymentBase::\$home\_path**

*string* = [\[line 28\]](#)

**Specifies the home path on the remote server**

- **Access** protected

**DeploymentBase::\$release\_path**

*string* = [\[line 34\]](#)

**Specifies the release path on the remote server**

- **Access** protected

Constructor *void* function DeploymentBase::\_\_construct() [\[line 42\]](#)

**Constructor**

- **Access** public

**Class Get**  
[\[line 19\]](#)

**deplomyent script**

- **Package** PHPRum

- **Sub-Package** Deploy
- **Author** Darnell Shinbine

void function Get::execute(\$argc, \$argv) [[line 35](#)]

**Function Parameters:**

- *int* **\$argc** Number of command line arguments
- *array* **\$argv** Array of command line arguments

## execute the application

- **Access** protected

array function Get::getOptions(\$argc, \$argv) [[line 143](#)]

**Function Parameters:**

- *int* **\$argc** Number of command line arguments
- *array* **\$argv** Array of command line arguments

## retrieve command line options

- **Access** protected

void function Get::handleError(\$errno, \$errstr, \$errfile, \$errline) [[line 190](#)]

**Function Parameters:**

- *string* **\$errno** error code
- *string* **\$errstr** error description

- *string* **\$errfile** file
- *string* **\$errline** line no.

event triggered by an error in the application, can be overridden to provide error handling.

- **Access** protected

*void* function Get::handleException(\\$e, \$e) [*line* [177](#)]

**Function Parameters:**

- *\Exception* **\$e**
- *Exception* **\\$e**

event triggered by an uncaught Exception thrown in the application, can be overridden to provide error handling.

- **Access** protected

## Class SSHClientBase

[*line* [24](#)]

**Provides base functionality for connecting via SSH**

The DeploymentBase exposes 4 protected properties, these properties must be defined in the sub class

- **Package** PHPRum
- **Sub-Package** Deploy
- **Property** string \$server: Specifies the server name
- **Property** string \$port: Specifies the server port
- **Property** string \$username: Specifies the login username
- **Property** string \$password: Specifies the login password
- **Author** Darnell Shinbine
- **Abstract Element**

#### **SSHClientBase::\$password**

*string* = " [[line 48](#)]

**Specifies the login password**

- **Access** protected

#### **SSHClientBase::\$port**

*int* = 22 [[line 36](#)]

**Specifies the server port**

- **Access** protected

#### **SSHClientBase::\$server**

*string* = 'localhost' [[line 30](#)]



## Specifies the server name

- **Access** protected

### SSHClientBase::\$username

*string* = 'root' [*line* [42](#)]

## Specifies the login username

- **Access** protected

*void* function SSHClientBase::exec() [*line* [94](#)]

### **execute deployment script**

- **Access** public

*void* function SSHClientBase::put(\$local\_path, \$remote\_path, \$repository\_path, \$home\_path) [*line* [76](#)]

#### **Function Parameters:**

- *string* **\$repository\_path** local path
- *string* **\$home\_path** remote path
- **\$local\_path**
- **\$remote\_path**

## put file on remote server

- **Access** protected

*void* function SSHClientBase::run(\$cmd) [*line* [63](#)]

***Function Parameters:***

- *string* **\$cmd** command to run on remote server

**execute on remote server**

- **Access** protected

# dbtranslator.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# filetranslator.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# translatorbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# translatorexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class DBTranslator

[*line* [18](#)]

**Represents a message Translator using a Database**

- **Package** PHPRum
- **Sub-Package** I18N
- **Author** Darnell Shinbine

Constructor *void* function DBTranslator::\_\_construct([\$table = "], [\$dsn = "]) [*line* [39](#)]

**Function Parameters:**

- *string* **\$table** table name
- *string* **\$dsn** database connection string

### Constructor

- **Access** public

*void function DBTranslator::get(\$stringId, [\$default = ""]) [line [65](#)]*

**Function Parameters:**

- *string* **\$stringId** string id to translate
- *string* **\$default** string if not found

## get

- **Access** public

# Class FileTranslator

*[line [18](#)]*

**Represents a message Translator using a File**

- **Package** PHPRum
- **Sub-Package** I18N
- **Author** Darnell Shinbine

*Constructor void function FileTranslator::\_\_construct([\$file = "], \$flie) [line [38](#)]*

**Function Parameters:**

- *string* **\$flie** file name

- **\$file**

## Constructor

- **Access** public

*void* function FileTranslator::get(\$stringId, [\$default = ""]) [*line* [58](#)]

### **Function Parameters:**

- *string* **\$stringId** string id to translate
- *string* **\$default** string if not found

## get

- **Access** public

# Class TranslatorBase

[*line* [21](#)]

**Represents a message Translator**

- **Package** PHPRum
- **Sub-Package** I18N



- **Property** string \$lang: specifies the language
- **Property** string \$charset: specifies character set
- **Author** Darnell Shinbine
- **Abstract Element**

Constructor *void* function TranslatorBase::\_\_construct() [*line 41*]

### **Constructor**

- **Access** public

*void* function TranslatorBase::get(\$stringId, [\$default = ""]) [*line 119*]

#### **Function Parameters:**

- *string* **\$stringId** string id to translate
- *string* **\$default** string if not found

### **get**

- **Abstract Element**
- **Access** public

*void* function TranslatorBase::setCharset(\$charset) [*line 105*]

#### **Function Parameters:**

- *string* **\$charset** character set

### **set character set**

- **Access** public

*void* function TranslatorBase::setLang(\$lang) [*line* [93](#)]

***Function Parameters:***

- *string* **\$lang** language

## set language

- **Access** public

# Class TranslatorException

[*line* [19](#)]

## Represents a Translator Exception

- **Package** PHPRum
- **Sub-Package** I18N
- **Author** Darnell Shinbine

# directorynotfoundexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# directorynotwritableexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# file.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# fileloadexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# filenotfoundexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# filenotwritableexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# filestream.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# filesystem.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# folder.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# ioexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# socketstream.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# streambase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# httprequest.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# httpresponse.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# session.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class DirectoryNotFoundException

[line [18](#)]

**Represents a Directory Not Found Exception**

- **Package** PHPRum
- **Sub-Package** IO
- **Author** Darnell Shinbine

## Class DirectoryNotWritableException

[line [18](#)]

**Represents a Directory Not Writable Exception**

- **Package** PHPRum
- **Sub-Package** IO
- **Author** Darnell Shinbine

## Class File

[line [25](#)]

**Represents a file on the file system**

- **Package** PHPRum
- **Sub-Package** IO
- **Property** string \$name: name of file
- **Property** string \$path: path to file
- **Property** string \$size: size of file
- **Property** int \$modified: specifies when file last modified
- **Property** int \$accessed: specifies when file last accessed
- **Property** int \$created: specifies when file created
- **Author** Darnell Shinbine

Constructor *void* function File::\_\_construct(\$filename) [line [109](#)]

**Function Parameters:**

- *string* **\$filename** Name of File

### Constructor

- **Access** public

*void function* File::changeFileMode([\$mode = 0755]) [*line* [211](#)]

***Function Parameters:***

- *string* **\$mode** chmod

## change file mode

- **Access** public

*void function* File::copy([\$dest = ""]) [*line* [133](#)]

***Function Parameters:***

- *string* **\$dest** path to destination

## copy file to destination

- **Access** public

*void function* File::delete() [*line* [171](#)]

## delete file

- **Access** public

*void* function File::move(\$dest) [*line* [152](#)]

**Function Parameters:**

- *string* **\$dest** path to destination

## move file to destination

- **Access** public

*string* function File::readLastLine(\$\_file) [*line* [230](#)]

**Function Parameters:**

- *string* **\$\_file** path to file to read

## returns the last line of text file **\*\*without\*\*** reading the entire file

- **Access** public

*void* function File::rename(\$newname) [*line* [191](#)]

**Function Parameters:**

- *string* **\$newname** new name

## rename file

- **Access** public

## Class FileLoadException

[line [18](#)]

**Represents a File Load Exception**

- **Package** PHPRum
- **Sub-Package** IO
- **Author** Darnell Shinbine

## Class FileNotFoundException

[line [18](#)]

**Represents a File Not Found Exception**

- **Package** PHPRum
- **Sub-Package** IO
- **Author** Darnell Shinbine

## Class FileNotWritableException

[line 18]

## Represents a File Not Writable Exception

- **Package** PHPRum
- **Sub-Package** IO
- **Author** Darnell Shinbine

## Class FileStream

[line 18]

Provides a way to stream data to and from a file

- **Package** PHPRum
- **Sub-Package** IO
- **Author** Darnell Shinbine

*bool* function FileStream::open(\$path, [\$mode = 'ab+']) [line 53]

### **Function Parameters:**

- *string* **\$path** path to file
- *int* **\$mode** a = read/write, w = write, r = readonly

**opens a text or binary file for input output**

- **Access** public

# Class FileSystem

[line [18](#)]

Provides access to the file system

- **Package** PHPRum
- **Sub-Package** IO
- **Author** Darnell Shinbine

void function FileSystem::changeFileMode(\$glob, [\$mode = 0755]) [line [212](#)]

## **Function Parameters:**

- *string* **\$glob** if string, must be a file name (foo.txt),  
(\* .txt), or directory name. if glob pattern
- **\$mode**

## Change the file mode (chmod)

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

void function FileSystem::copy(\$src, [\$dest = "]) [line [27](#)]

## **Function Parameters:**

- *string* **\$src** Source path

- *string* **\$dest** Destination path

## Recursively copy a file or folder and its contents

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

*void* function FileSystem::delete(\$glob) [*line* [157](#)]

### **Function Parameters:**

- *string* **\$glob** if string, must be a file name (foo.txt), if glob pattern  
(\*.txt), or directory name.

## Recursively delete a file or folder and its contents using wild cards

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

*void* function FileSystem::move(\$src, \$dest) [*line* [100](#)]

### **Function Parameters:**

- *string* **\$src** Source path
- *string* **\$dest** Destination path

## Recursively move a file or folder and its contents



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

*void function* FileSystem::removeDirectory(\$directory) [*line* [242](#)]

***Function Parameters:***

- *string* **\$directory** Path of directory

## Recursively delete a dir and its contents

- **Static**

*void function* FileSystem::rename(\$old, \$new) [*line* [185](#)]

***Function Parameters:***

- *string* **\$old** Old filename
- *string* **\$new** New filename

## Rename a file or folder to another location

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

# Class Folder

[[line 26](#)]

**Represents a folder on the file system**

- **Package** PHPRum
- **Sub-Package** IO
- **Property** string \$name: name of folder
- **Property** string \$path: path to folder
- **Property** array \$files: list of files
- **Property** array \$folders: list of sub folders
- **Property** int \$modified: specifies when folder last modified
- **Property** int \$accessed: specifies when folder last accessed
- **Property** int \$created: specifies when folder created
- **Author** Darnell Shinbine

Constructor *void* function Folder::\_\_construct(\$dirname) [[line 120](#)]

**Function Parameters:**

- *string* **\$dirname** Name of Directory

## Constructor

- **Access** public

*void* function Folder::changeFileMode([\$mode = 0755]) [[line 241](#)]

**Function Parameters:**

- *string* **\$mode** chmod

## change file mode

- **Access** public

*void function* Folder::copy([\$dest = "]) [*line* [163](#)]

### **Function Parameters:**

- *string* **\$dest** path to destination

## copy folder to destination

- **Access** public

*void function* Folder::delete() [*line* [201](#)]

### **delete folder**

- **Access** public

*void function* Folder::move(\$dest) [*line* [182](#)]

### **Function Parameters:**

- *string* **\$dest** path to destination

## move folder to destination

- **Access** public

*void* function Folder::rename(\$newname) [*line* [221](#)]

**Function Parameters:**

- *string* **\$newname** new name

## rename folder

- **Access** public

# Class HTTPRequest

[*line* [23](#)]

**Provides a way to access HTTP Request information from the client**

- **Package** PHPRum
- **Sub-Package** IO
- **Property** array \$post: HTTP POST request variables
- **Property** array \$get: HTTP GET request variables
- **Property** array \$request: HTTP GET/POST request variables
- **Property** array \$cookies: HTTP Cookies
- **Author** Darnell Shinbine

## HttpRequest::\$cookie

*array* = null [[line 41](#)]

### var to hold cookie vars

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

## HttpRequest::\$get

*array* = null [[line 29](#)]

### var to hold get vars

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

## HttpRequest::\$post

*array* = null [[line 35](#)]

### var to hold post vars

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

## HttpRequest::\$request

*array* = null [[line 47](#)]

## var to hold request vars

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

Constructor *void* function HTTPRequest::\_\_construct() [[line 57](#)]

### **Constructor**

Read values from request, clean up variables, and merge get and post requests.

- **Access** public

*string* function HTTPRequest::getQueryString() [[line 206](#)]

### **return the URL**

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

*string* function HTTPRequest::getRawPostData() [[line 226](#)]

### **return raw post data**

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

*array* function HTTPRequest::getRequestHeaders() [[line 256](#)]

## return request headers

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

*string* function HTTPRequest::getRequestMethod() [*line [246](#)*]

## return the request method

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

*string* function HTTPRequest::getRequestReferer() [*line [236](#)*]

## return the request method

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

*string* function HTTPRequest::getRequestURI() [*line [216](#)*]

## return the URI

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

# Class HTTPResponse

[[line 23](#)]

Provides a way to send HTTP response information to the client

- **Package** PHPRum
- **Sub-Package** IO
- **Property** int \$statusCode: code HTTP response status code
- **Property** string \$status: HTTP response status
- **Property** array \$headers: HTTP response headers
- **Property** string \$contents: HTTP response
- **Author** Darnell Shinbine

Constructor *void* function HTTPResponse::\_\_construct() [[line 55](#)]

## Constructor

- **Access** public

*void* function HTTPResponse::addHeader(\$header) [[line 140](#)]

## Function Parameters:

- *string* **\$header** header information

**adds header to output stream**



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

*void* function `HTTPResponse::clear()` [[line 195](#)]  
**clear all content from output buffer**

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

*void* function `HTTPResponse::end()` [[line 213](#)]  
**send response to client and stop execution**

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

*void* function `HTTPResponse::flush()` [[line 177](#)]  
**send all content to client**

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

*string* function `HTTPResponse::getResponseContent()` [[line 233](#)]  
**return response content**

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

*StringCollection* function `HTTPResponse::getResponseHeaders()` [*line* [286](#)]  
**return response headers**

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

*void* function `HTTPResponse::redirect($location, [$terminate = true])` [*line* [270](#)]  
**Function Parameters:**

- *string* **\$location** the URL to redirect to
- *bool* **\$terminate** specifies whether to stop execution

## redirect to new URL

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

*void* function `HTTPResponse::setCookie($name, [$value = ''], [$expires = 0], [$path = '/'], [$domain = null], [$secure = false])` [*line* [249](#)]  
**Function Parameters:**

- *string* **\$name** the name of the cookie
- *string* **\$value** the value of the cookie
- *int* **\$expires** timestamp that the cookie will expire on
- *string* **\$path** the path on the server that the cookie will be available on

- *string* **\$domain** the domain that the cookie is available on
- *bool* **\$secure** requires the cookie be sent over secure HTTPS

## set cookie

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

*void* function HTTPResponse::write(\$content) [*line* [159](#)]

### **Function Parameters:**

- *string* **\$content** information to send

## sends information to an output stream

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

# Class IOException

[*line* [19](#)]

**Represents an I/O Exception**

- **Package** PHPRum
- **Sub-Package** IO
- **Author** Darnell Shinbine

## Class Session

[[line 21](#)]

**Provides a way to access the persistant session**

- **Package** PHPRum
- **Sub-Package** IO
- **Property** string \$sessionId: Session Id
- **Property** string \$sessionName: Session Name
- **Author** Darnell Shinbine

Constructor *void* function Session::\_\_construct() [[line 47](#)]

**Read values from request, clean up variables, and merge get and post requests.**

- **Access** public

*void* function Session::close() [[line 254](#)]

**close the session**

- **Access** public

*void* function Session::destroy() [[line 271](#)]

**destroy the session**

- **Access** public

*array* function Session::getSessionData() [[line 327](#)]

**return session array**

- **Access** public

*void* function Session::read(\$sessId) [[line 211](#)]

**Function Parameters:**

- *string* **\$sessId** new session id

**initialize and read session data once started (implicit)**

- **Access** public

*void* function Session::regenerateId([\$delete = false]) [[line 291](#)]

**Function Parameters:**

- *bool* **\$delete** specifies whether to delete the old associated session file or not

**regenerate the session id**

- **Access** public

*void* function Session::removeAll() [*line* [316](#)]  
**remove all session keys**

- **Access** public

*void* function Session::start([\$sessId = null]) [*line* [181](#)]  
**Function Parameters:**

- *string* **\$sessId** new session id

**This method will start a new session or continue an existing session.**  
You can optionally set the session id

- **Access** public

*void* function Session::write() [*line* [230](#)]  
**write and close the session**

- **Access** public

# Class SocketStream

[line [18](#)]

Provides a way to stream data to and from a socket

- **Package** PHPRum
- **Sub-Package** IO
- **Author** Darnell Shinbine

*bool* function SocketStream::open(\$host, [\$port = 80], [\$timeout = 30]) [line [41](#)]

**Function Parameters:**

- *string* **\$host** host
- *int* **\$port** port
- *int* **\$timeout** timeout is seconds

opens a socket for input output

- **Access** public

# Class StreamBase

[line [18](#)]

Provides base functionality for streaming

- **Package** PHPRum
- **Sub-Package** IO
- **Author** Darnell Shinbine
- **Abstract Element**

## **StreamBase::\$handle**

*resource* = null [*line* [24](#)]

**handle to file**

- **Access** protected

*void* function StreamBase::close() [*line* [44](#)]

**closes an open connection**

- **Access** public

*bool* function StreamBase::eos() [*line* [312](#)]

**Returns whether the current position is at the end of the stream or not**

- **Access** public

*void* function StreamBase::flush() [*line* [330](#)]

**Sends the contents of the Stream buffer to the associated underlying object**



- **Access** public

*int* function StreamBase::getPosition() [*line* [243](#)]

**Returns the current position from the beginning of a Stream object**

- **Access** public

*int* function StreamBase::length() [*line* [192](#)]

**Returns the size of the data**

- **Access** public

*string* function StreamBase::read([\$bytes = 0]) [*line* [63](#)]

**Function Parameters:**

- *int* **\$bytes** no. of bytes to read

**Reads the entire stream or a specified number of bytes from a binary Stream object**

- **Access** public

*string* function StreamBase::readln(\$bytes) [*line* [103](#)]

**Function Parameters:**

- *int* **\$bytes** no. of bytes to read

**Reads one line from a TextStream file and returns the result**

- **Access** public

*void* function StreamBase::setBOS() [[line 270](#)]

**Sets the current position to be the end of the stream (EOS)**

- **Access** public

*void* function StreamBase::setEOS() [[line 291](#)]

**Sets the current position to be the end of the stream (EOS)**

- **Access** public

*void* function StreamBase::setPosition(\$offset) [[line 215](#)]

**Function Parameters:**

- *int* **\$offset** number of bytes from beggining of file

**Sets the current position from the beginning of a Stream object**

- **Access** public

*int* function StreamBase::write(\$raw) [*line* [132](#)]

**Function Parameters:**

- *string* **\$raw** raw binary data or text string

**Writes binary data to a binary Stream object**

- **Access** public

*int* function StreamBase::writeln(\$line, \$string) [*line* [167](#)]

**Function Parameters:**

- *string* **\$string** string to write
- **\$line**

**Writes a specified number of new-line character to a TextStream file**

- **Access** public

# dblogger.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# filelogger.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# loggerexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class DBLogger

[line [22](#)]

**Provides access to read/write log messages**

- **Package** PHPRum
- **Sub-Package** Logger
- **Property** string \$path: path to log file
- **Property** string \$file: name of log file
- **Property** int \$maxFileSize: max size of log file in KB, 0 for infinite
- **Author** Darnell Shinbine

Constructor *void* function DBLogger::\_\_construct([\$table = \_\_LOGS\_TABLENAME\_\_], [\$dsn = ""]) [line [43](#)]

### **Function Parameters:**

- *string* **\$table** table name
- *string* **\$dsn** database connection string

## Constructor

- **Access** public

*void* function DBLogger::flush(\$category) [[line 133](#)]

### **Function Parameters:**

- *string* **\$category** message category

**This method flushes all messages**

- **Access** public

*void* function DBLogger::log(\$message, \$category) [[line 65](#)]

### **Function Parameters:**

- *string* **\$message** message to log
- *string* **\$category** message category

**This method writes a log message to memory**

- **Access** public

*array* function DBLogger::logs(\$category) [[line 117](#)]

### **Function Parameters:**

- *string* **\$category** message category

**This method retrieves all log messages**

- **Access** public

## Class FileLogger

[line [22](#)]

**Provides access to read/write log messages**

- **Package** PHPRum
- **Sub-Package** Logger
- **Property** string \$path: path to log file
- **Property** string \$file: name of log file
- **Property** int \$maxFileSize: max size of log file in KB, 0 for infinite
- **Author** Darnell Shinbine

Constructor *void* function FileLogger::\_\_construct([\$path = \_\_LOG\_PATH\_\_], [\$maxFileSize = 0]) [line [44](#)]

**Function Parameters:**

- *string* **\$path** path of log files
- *string* **\$maxFileSize** max file size

### Constructor



- **Access** public

*void* function FileLogger::flush(\$category) [*line* [166](#)]

**Function Parameters:**

- *string* **\$category** message category

**This method flushes all messages**

- **Access** public

*void* function FileLogger::log(\$message, \$category) [*line* [107](#)]

**Function Parameters:**

- *string* **\$message** message to log
- *string* **\$category** message category

**This method writes a log message to memory**

- **Access** public

*array* function FileLogger::logs(\$category) [*line* [138](#)]

**Function Parameters:**

- *string* **\$category** message category

**This method retrieves all log messages**

- **Access** public

## Class LoggerException

*[line [19](#)]*

**Represents a Logger Exception**

- **Package** PHPRum
- **Sub-Package** Logger
- **Author** Darnell Shinbine

# imapclient.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# mailmessage.class.php

- **Package** PHPRum
- **Filesource** [Source Code for this file](#)
- **License** see

# phpmailclient.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# sendmailclient.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# smtpclient.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

require\_once \_\_LIB\_PATH\_\_.'/smtp/class.smtp.php' [*line 13*]

**include lib**

## Class IMailClient

[*line 18*]

**Provides the interface for Mail Clients**

- **Package** PHPRum
- **Sub-Package** Mail
- **Author** Darnell Shinbine

*void* function IMailClient::send(\$message) [*line 26*]

**Function Parameters:**

- [MailMessage](#) **\$message** message to send

**sends a single email to all addresses in message**

- **Access** public

## Class MailMessage

[line [29](#)]

**Represents an E-Mail Message**

- **Package** PHPRum
- **Sub-Package** Mail
- **Property** string \$to: Recipient email address
- **Property** string \$from: Sender's email address
- **Property** string \$subject: Message subject
- **Property** string \$body: Message body
- **Property** string \$charset: Message character set
- **Property** string \$contentType: Message content-type
- **Property** string \$encoding: Message encoding
- **Property** bool \$html: Specifies whether to send as html
- **Property** array \$cc: array of email addresses to be copied
- **Property** array \$bcc: array of email addresses to be blind copied

Constructor *void* function MailMessage::\_\_construct() [line [109](#)]

**Constructor**



- **Access** public

*void function MailMessage::addBcc(\$bcc) [line [340](#)]*

***Function Parameters:***

- *string* **\$bcc** email address

## **add email address to blind copy**

- **Access** public

*void function MailMessage::addCc(\$cc) [line [321](#)]*

***Function Parameters:***

- *string* **\$cc** email address

## **add email address to copy**

- **Access** public

*void function MailMessage::attach(\$file, [\$mimetype = 'text/plain'], [\$filename = ''], [\$inline = false], [\$alternate = false], [\$encoding = '7bit'], \$disposition) [line [364](#)]*

***Function Parameters:***

- *string* **\$file** file path or content
- *string* **\$mimetype** attachment mimetype
- *string* **\$filename** optional filename (auto detected if adding by path)

- *string* **\$disposition** specifies whether the attachment disposition is inline, default is false
- *string* **\$alternate** specifies whether the attachment is part of a multipart/alternate message, default is false
- *string* **\$encoding** optional encoding, default is 7bit
- **\$inline**

## add an attachment to the mail message

- **Access** public

*string* function MailMessage::getContent() [[line 424](#)]  
**get message content**

- **Access** public

*string* function MailMessage::getHeaders() [[line 392](#)]  
**get message headers**

- **Access** public

*void* function MailMessage::setBody(\$body) [[line 273](#)]  
**Function Parameters:**

- *string* **\$body** message body

## set message body

- **Access** public

*void* function MailMessage::setCharset(\$charset) [[line 285](#)]

### **Function Parameters:**

- *string* **\$charset** character set

## set character set

- **Access** public

*void* function MailMessage::setContentType(\$contentType) [[line 297](#)]

### **Function Parameters:**

- *string* **\$contentType** content type

## set content type

- **Access** public

*void* function MailMessage::setEncoding(\$encoding) [[line 309](#)]

### **Function Parameters:**

- *string* **\$encoding** encoding

## set character encoding

- **Access** public

*void* function MailMessage::setFrom(\$from) [[line 242](#)]

### **Function Parameters:**

- *string* **\$from** sender's email address

## set sender's email address

- **Access** public

*void* function MailMessage::setSubject(\$subject) [[line 261](#)]

### **Function Parameters:**

- *string* **\$subject** message subject

## set message subject

- **Access** public

*void* function MailMessage::setTo(\$to) [[line 223](#)]

### **Function Parameters:**

- *string* **\$to** recipient email address

## set recipient email address

- **Access** public

## Class PHPMailClient

[line [18](#)]

Handles email sending via PHP mail() function

- **Package** PHPRum
- **Sub-Package** Mail
- **Author** Darnell Shinbine

void function PHPMailClient::send(\$message) [line [26](#)]

**Function Parameters:**

- [MailMessage](#) **\$message** message to send

sends a single email to all addresses in message

- **Access** public

# Class SendMailClient

[[line 20](#)]

Handles email sending via the \$sendmail program

- **Package** PHPRum
- **Sub-Package** Mail
- **Property** string \$path: specifies the path to the \$sendmail program
- **Author** Darnell Shinbine

Constructor *void* function SendMailClient::\_\_construct([\$path = ""]) [[line 33](#)]

**Function Parameters:**

- *string* **\$path** specifies the path to the \$sendmail program

## Constructor

- **Access** public

*void* function SendMailClient::send(\$message) [[line 86](#)]

**Function Parameters:**

- [MailMessage](#) **\$message** message to send

sends a single email to all addresses in message

- **Access** public

## Class SMTPClient

[*line* [30](#)]

Handles email sending via smtp client

- **Package** PHPRum
- **Sub-Package** Mail
- **Property** string \$host: specifies the smtp host
- **Property** int \$port: specifies the smtp port
- **Property** int \$timeout: specifies the smtp connection timeout
- **Property** bool \$keepAlive: specifies whether to keep the connection alive
- **Property** string \$helo: specifies the smtp helo message
- **Property** string \$authUsername: specifies the smtp authentication username
- **Property** string \$authPassword: specifies the smtp authentication password
- **Author** Darnell Shinbine

Constructor *void* function SMTPClient::\_\_construct([\$host = ""]) [*line* [85](#)]

**Function Parameters:**

- *string* **\$host** specifies the smtp host

### Constructor

- **Access** public

*void* function SMTPClient::close() [[line 239](#)]

**Closes the active SMTP session if one exists.**

- **Access** public

*void* function SMTPClient::open() [[line 184](#)]

**Initiates a connection to an SMTP server operation failed.**

- **Access** public

*void* function SMTPClient::send(\$message) [[line 259](#)]

**Function Parameters:**

- [MailMessage](#) **\$message** message to send

**sends a single email to all addresses in message**

- **Access** public



# makebase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# controller.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# formcontroller.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# gridviewController.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# jsonwebservice.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# listcontroller.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# model.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# pagecontroller.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# restwebservice.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# soapwebservice.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class Controller

[*line 17*]

**Provides functionality to generate controller/view files**

- **Package** PHPRum
- **Sub-Package** Make
- **Author** Darnell Shinbine

### Controller::ControllerNamespace

= '\\Controllers' [*line 23*]

### Default namespace

*void* function Controller::make(\$target, [\$options = array()]) [*line 33*]

#### **Function Parameters:**

- *string* **\$target** target
- *array* **\$options** options

**make**

- **Access** public

## Class FormController

*[line [17](#)]*

**Provides functionality to generate controller/view files**

- **Package** PHPRum
- **Sub-Package** Make
- **Author** Darnell Shinbine

**FormController::ControllerNamespace**

= '\\Controllers' *[line [23](#)]*

**Default namespace**

**FormController::DefaultBaseController**

= '\\ApplicationController' *[line [35](#)]*

**DefaultBaseController**

**FormController::ModelNamespace**

= '\\Models' *[line [29](#)]*

**Default namespace**

`void function FormController::make($target, [$options = array()]) [line 45]`

**Function Parameters:**

- *string* **\$target** target
- *array* **\$options** options

**make**

- **Access** public

## Class GridViewController

[line 17]

**Provides functionality to generate controller/view files**

- **Package** PHPRum
- **Sub-Package** Make
- **Author** Darnell Shinbine

### GridViewController::ControllerNamespace

= '\\Controllers' [line 23]

### Default namespace

### GridViewController::DefaultBaseController

= '\\ApplicationController' [[line 34](#)]

## DefaultBaseController

### GridViewController::ModelNamespace

= '\\Models' [[line 28](#)]

## Default namespace

void function GridViewController::make(\$target, [\$options = array()]) [[line 44](#)]

### **Function Parameters:**

- *string* **\$target** target
- *array* **\$options** options

## make

- **Access** public

# Class JSONWebService

[[line 17](#)]

**Provides functionality to generate controller/view files**

- **Package** PHPRum
- **Sub-Package** Make
- **Author** Darnell Shinbine

## JSONWebService::ControllerNamespace

= '\\Controllers' [[line 23](#)]

## Default namespace

void function JSONWebService::make(\$target, [\$options = array()]) [[line 33](#)]

### Function Parameters:

- *string* **\$target** target
- *array* **\$options** options

## make

- **Access** public

# Class ListController

[[line 17](#)]

Provides functionality to generate controller/view files

- **Package** PHPRum
- **Sub-Package** Make
- **Author** Darnell Shinbine

## ListController::ControllerNamespace

= '\\Controllers' [[line 23](#)]

## Default namespace

**ListController::DefaultBaseController**

= '\\ApplicationController' [[line 34](#)]

## DefaultBaseController

**ListController::ModelNamespace**

= '\\Models' [[line 28](#)]

## Default namespace

*void* function ListController::make(\$target, [\$options = array()]) [[line 44](#)]

### **Function Parameters:**

- *string* **\$target** target
- *array* **\$options** options

## make

- **Access** public

# Class MakeBase

[[line 18](#)]

**Provides the interface for Make scripts**

- **Package** PHPRum
- **Sub-Package** Make

- **Author** Darnell Shinbine
- **Abstract Element**

void function MakeBase::export(\$path, \$contents) [[line 37](#)]

**Function Parameters:**

- *string* **\$path** path
- *string* **\$contents** contents

## export

- **Access** protected

void function MakeBase::make(\$target, [\$options = array()]) [[line 27](#)]

**Function Parameters:**

- *string* **\$target** target
- *array* **\$options** options

## make

- **Abstract Element**
- **Access** public



# Class Model

[[line 17](#)]

Provides functionality to generate a model file

- **Package** PHPRum
- **Sub-Package** Make
- **Author** Darnell Shinbine

**Model::ModelNamespace**

= '\\Models' [[line 23](#)]

**Default namespace**

*void* function Model::make(\$target, [\$options = array()]) [[line 33](#)]

**Function Parameters:**

- *string* **\$target** target
- *array* **\$options** options

**make**

- **Access** public

# Class PageController

[[line 17](#)]

## Provides functionality to generate controller/view files

- **Package** PHPRum
- **Sub-Package** Make
- **Author** Darnell Shinbine

### PageController::ControllerNamespace

= '\\Controllers' [[line 23](#)]

### Default namespace

### PageController::DefaultBaseController

= '\\ApplicationController' [[line 29](#)]

### DefaultBaseController

*void* function PageController::make(\$target, [\$options = array()]) [[line 39](#)]

#### **Function Parameters:**

- *string* **\$target** target
- *array* **\$options** options

### make

- **Access** public

# Class RESTWebService

[\[line 17\]](#)

Provides functionality to generate controller/view files

- **Package** PHPRum
- **Sub-Package** Make
- **Author** Darnell Shinbine

## RESTWebService::ControllerNamespace

= '\\Controllers' [\[line 23\]](#)

### Default namespace

void function RESTWebService::make(\$target, [\$options = array()]) [\[line 33\]](#)

#### **Function Parameters:**

- *string* **\$target** target
- *array* **\$options** options

### make

- **Access** public

# Class SOAPWebService

[\[line 17\]](#)

Provides functionality to generate controller/view files

- **Package** PHPRum
- **Sub-Package** Make
- **Author** Darnell Shinbine

## SOAPWebService::ControllerNamespace

= '\\Controllers' [[line 23](#)]

## Default namespace

*void* function SOAPWebService::make(\$target, [\$options = array()]) [[line 33](#)]

### **Function Parameters:**

- *string* **\$target** target
- *array* **\$options** options

## make

- **Access** public

# migrate.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# migrationbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# migrationcompare.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# migrations.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# test.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class Migrate

[line [19](#)]

deplomyent script

- **Package** PHPRum
- **Sub-Package** Migrate
- **Author** Darnell Shinbine

*void* function Migrate::execute(\$argc, \$argv) [line [29](#)]

**Function Parameters:**

- *int* **\$argc** Number of command line arguments
- *array* **\$argv** Array of command line arguments

execute the application

- **Access** protected

array function Migrate::getOptions(\$argc, \$argv) [[line 72](#)]

**Function Parameters:**

- *int* **\$argc** Number of command line arguments
- *array* **\$argv** Array of command line arguments

## retrieve command line options

- **Access** protected

void function Migrate::handleError(\$errno, \$errstr, \$errfile, \$errline) [[line 119](#)]

**Function Parameters:**

- *string* **\$errno** error code
- *string* **\$errstr** error description
- *string* **\$errfile** file
- *string* **\$errline** line no.

event triggered by an error in the application, can be overridden to provide error handling.

- **Access** protected

void function Migrate::handleException(\Exception \$e, \$e) [[line 106](#)]

**Function Parameters:**

- `\Exception $e`
- `Exception $e`

event triggered by an uncaught Exception thrown in the application, can be overridden to provide error handling.

- **Access** protected

## Class MigrationBase

*[line 18]*

Provides base functionality for database migration

- **Package** PHPRum
- **Sub-Package** Migrate
- **Author** Darnell Shinbine
- **Abstract Element**

**MigrationBase::\$db**

*SQLDataAdapter = [line 24]*

**database**

- **Access** protected

## MigrationBase::\$version

*int* = [\[line 30\]](#)

## version

- **Access** public

Constructor *void* function MigrationBase::\_\_construct() [\[line 36\]](#)

## Constructor

- **Access** public

*void* function MigrationBase::down() [\[line 51\]](#)

## down migration

- **Abstract Element**
- **Access** public

*void* function MigrationBase::up() [\[line 45\]](#)

## up migration

- **Abstract Element**

- **Access** public

## Class MigrationCompare

[line [18](#)]

**Provides migration sorting**

- **Package** PHPRum
- **Sub-Package** Migrate
- **Author** Darnell Shinbine

*int* function MigrationCompare::compareVersion(\$a, \$b) [line [27](#)]

**Function Parameters:**

- *real* **\$a** ints a
- *real* **\$b** ints b

**method compares version numbers**

- **Access** public

## Class Migrations

[line [18](#)]

## Provides base functionality for database migration

- **Package** PHPRum
- **Sub-Package** Migrate
- **Author** Darnell Shinbine

*void* function Migrations::downgrade([\$toVersion = null]) [line [59](#)]

### **Function Parameters:**

- *real* **\$toVersion** optional version, defaults to previous version

## downgrade database

- **Access** public

*real* function Migrations::getCurrentVersion() [line [113](#)]

## get database version

- **Access** public

*void* function Migrations::to(\$toVersion) [line [94](#)]

### **Function Parameters:**

- *real* **\$toVersion** version

## set database version

- **Access** public

*void* function Migrations::upgrade([\$toVersion = null]) [*line* [31](#)]

**Function Parameters:**

- *real* **\$toVersion** optional version, defaults to latest version

## upgrade database

- **Access** public

## Class Test

[*line* [19](#)]

## deplomyent script

- **Package** PHPRum
- **Sub-Package** Migrate
- **Author** Darnell Shinbine

*void* function Test::execute(\$argc, \$argv) [*line* [29](#)]

**Function Parameters:**

- *int* **\$argc** Number of command line arguments
- *array* **\$argv** Array of command line arguments

## execute the application

- **Access** protected

*void* function Test::handleError(\$errno, \$errstr, \$errfile, \$errline) [*line* [88](#)]

### **Function Parameters:**

- *string* **\$errno** error code
- *string* **\$errstr** error description
- *string* **\$errfile** file
- *string* **\$errline** line no.

event triggered by an error in the application, can be overridden to provide error handling.

- **Access** protected

*void* function Test::handleException(\Exception \$e, \$e) [*line* [75](#)]

### **Function Parameters:**

- *\Exception* **\$e**
- *Exception* **\\$e**

event triggered by an uncaught Exception thrown in the application, can be overridden to provide error handling.



- **Access** protected

# authentication.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# authenticationstatus.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# basicauthentication.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# credentialbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# formsauthentication.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# roles.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# tablecredential.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# usercredential.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# webserviceauthentication.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class Authentication

[line [18](#)]

**Provides application wide authentication**

- **Package** PHPRum
- **Sub-Package** Security
- **Author** Darnell Shinbine

**Authentication::\$identity**

*string* = [line [24](#)]

**authenticated id**

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

## Authentication::\$isProtected

*bool* = null [[line 30](#)]

**specifies whether the current controller is protected**

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

*AuthenticationStatus* function Authentication::authenticate(\$username, \$password) [[line 48](#)]

### **Function Parameters:**

- *string* **\$username** specifies username
- *string* **\$password** specifies password

**authenticate user/password based on credentials (does not set cookie)**

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

*bool* function Authentication::authenticated() [[line 117](#)]

**returns true if user authenticated**

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

*bool* function Authentication::authorized() [[line 162](#)]

**returns true if user authenticated**

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

*string* function Authentication::generateHash(\$passwordFormat, \$passwordToEncrypt, \$salt) [[line 298](#)]

**Function Parameters:**

- *string* **\$passwordFormat** password format
- *string* **\$passwordToEncrypt** password to encrypt
- *string* **\$salt** salt

**generates a hash based on the password**

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

*string* function Authentication::generateSalt([\$len = 15]) [[line 325](#)]

**Function Parameters:**

- *string* **\$len** length of salt

**generates a salt**

- **Static**

- **Access** public

*string* function Authentication::getAuthMethod() [[line 268](#)]  
**returns authentication method**

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

*bool* function Authentication::isProtected(\$controllerId) [[line 279](#)]  
**Function Parameters:**

- *string* **\$controllerId** id of controller

**returns true if controller is protected (requires authentication to process)**

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

*void* function Authentication::redirectToLogin() [[line 242](#)]  
**redirect user to login**

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

*void* function Authentication::signout() [[line 92](#)]  
**perform sign out (does not end session)**

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

## Class AuthenticationStatus

*[line 18]*

**Provides application wide authentication**

- **Package** PHPRum
- **Sub-Package** Security
- **Author** Darnell Shinbine

Constructor *void* function AuthenticationStatus::\_\_construct([\$invalidCredentials = false], [\$disabled = false], [\$lockedOut = false]) *[line 50]*

**Function Parameters:**

- **\$invalidCredentials**
- **\$disabled**
- **\$lockedOut**

???

- **Var** / Constructor

- **Access** public

*bool* function AuthenticationStatus::authenticated() [*line* [63](#)]  
**returns true if the account is authenticated**

- **Access** public

*bool* function AuthenticationStatus::disabled() [*line* [85](#)]  
**returns true if the account is locked out**

- **Access** public

*bool* function AuthenticationStatus::invalidCredentials() [*line* [74](#)]  
**returns true if invalid credentials supplied**

- **Access** public

*bool* function AuthenticationStatus::lockedOut() [*line* [96](#)]  
**returns true if the account is locked out**

- **Access** public

# Class BasicAuthentication

[[line 18](#)]

**Provides application wide authentication using Basic HTTP Headers**

- **Package** PHPRum
- **Sub-Package** Security
- **Author** Darnell Shinbine

*bool* function BasicAuthentication::authenticated() [[line 25](#)]

**returns true if user authenticated**

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

*string* function BasicAuthentication::getAuthUser() [[line 66](#)]

**gets auth user**

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

*bool* function BasicAuthentication::isAuthUserSet() [[line 84](#)]

**returns true if auth is set**

- **Static**



- **Access** public

*void* function BasicAuthentication::sendAuthHeaders() [*line* [44](#)]  
**send auth http headers**

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

*void* function BasicAuthentication::signout() [*line* [95](#)]  
**perform sign out**

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

## Class CredentialBase

[*line* [18](#)]

**Provides application wide authentication**

- **Package** PHPRum
- **Sub-Package** Security
- **Author** Darnell Shinbine
- **Abstract Element**

## CredentialBase::\$credential

*array* = array() [*line 24*]

### credential data

- **Access** protected

Constructor *void* function CredentialBase::\_\_construct(\$credential) [*line 31*]

#### **Function Parameters:**

- *array* **\$credential** credential data

### Constructor

- **Access** public

*AuthenticationStatus* function CredentialBase::authenticate(\$username, \$password) [*line 44*]

#### **Function Parameters:**

- *string* **\$username** specifies username
- *string* **\$password** specifies password

### authenticates password based on the credential

- **Abstract Element**
- **Access** public

*bool* function CredentialBase::comparePassword(\$encryptedPassword, \$passwordToCompare, \$salt) [*line* [54](#)]

**Function Parameters:**

- *type* **\$encryptedPassword**
- *type* **\$passwordToCompare**
- **\$salt**

**compare passwords, return true on success**

- **Abstract Element**
- **Access** public

## Class FormsAuthentication

[*line* [18](#)]

**Provides application wide authentication using Forms**

- **Package** PHPRum
- **Sub-Package** Security
- **Author** Darnell Shinbine

*bool* function FormsAuthentication::authenticated() [*line* [25](#)]

**returns true if user authenticated**

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

*void* function FormsAuthentication::authenticateSecret(\$uid, \$secret) [*line* [89](#)]

**Function Parameters:**

- *string* **\$uid** unique value representing user
- *string* **\$secret** secret

**authenticate secret**

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

*string* function FormsAuthentication::getAuthCookie() [*line* [134](#)]

**gets auth cookie**

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

*string* function FormsAuthentication::getAuthSecret() [*line* [156](#)]

**gets auth secret**

- **Static**

- **Access** public

*bool* function FormsAuthentication::isAuthCookieSet() [*line* [178](#)]

**returns true if auth cookie is set**

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

*void* function FormsAuthentication::redirectFromLoginPage(\$uid, [\$permanent = false]) [*line* [42](#)]

**Function Parameters:**

- *string* **\$uid** unique value representing user
- *string* **\$permanent** specifies whether cookie is permanent

**sets auth cookie and redirects to original requested resource**

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

*void* function FormsAuthentication::redirectToLoginPage() [*line* [64](#)]

**redirects to login page**

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

`void function FormsAuthentication::setAuthCookie($uid, [$permanent = false], [$expires = 31536000])` [[line 110](#)]

**Function Parameters:**

- *string* **\$uid** unique value representing user
- *string* **\$permanent** specifies whether cookie is permanent
- *int* **\$expires** specifies time in seconds for cookie to expire, default 1 year

**sets auth cookie**

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

`void function FormsAuthentication::signout()` [[line 200](#)]

**perform sign out (does not end session)**

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

## Class Roles

[[line 18](#)]

**Provides application wide authentication**

- **Package** PHPRum

- **Sub-Package** Security
- **Author** Darnell Shinbine

*array* function Roles::getRolesForUser() [*line 64*]

**get roles for current user**

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

*bool* function Roles::isUserInRole(\$roleIdentifier) [*line 34*]

**Function Parameters:**

- *string* **\$roleIdentifier** role identifier

**returns true if current user is a member of the provide role**

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

*bool* function Roles::isUserInRoles(\$roleIdentifiers) [*line 46*]

**Function Parameters:**

- *array* **\$roleIdentifiers** array of role identifiers

**returns true if current user is a member of the provided roles**

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

## Class TableCredential [line [18](#)]

**Provides application wide authentication**

- **Package** PHPRum
- **Sub-Package** Security
- **Author** Darnell Shinbine

*AuthenticationStatus* function TableCredential::authenticate(\$username, \$password) [line [27](#)]

### **Function Parameters:**

- *string* **\$username** specifies username
- *string* **\$password** specifies password

**authenticates password based on the credential**

- **Access** public

*bool* function TableCredential::comparePassword(\$encryptedPassword, \$passwordToCompare, \$salt) [line [75](#)]

### **Function Parameters:**



- *type* **\$encryptedPassword**
- *type* **\$passwordToCompare**
- **\$salt**

**compare passwords, return true on success**

- **Access** public

*string* function TableCredential::generateHash(\$passwordToEncrypt, \$salt) [*line* [86](#)]

***Function Parameters:***

- **\$passwordToEncrypt**
- **\$salt**

**generate password hash**

- **Access** public

## Class UserCredential

[*line* [18](#)]

**Provides application wide authentication**

- **Package** PHPRum

- **Sub-Package** Security
- **Author** Darnell Shinbine

*AuthenticationStatus* function UserCredential::authenticate(\$username, \$password) [[line 27](#)]

**Function Parameters:**

- *string* **\$username** specifies username
- *string* **\$password** specifies password

**authenticates password based on the credential**

- **Access** public

*bool* function UserCredential::comparePassword(\$encryptedPassword, \$passwordToCompare, \$salt) [[line 64](#)]

**Function Parameters:**

- *type* **\$encryptedPassword**
- *type* **\$passwordToCompare**
- **\$salt**

**compare passwords, return true on success**

- **Access** public

# Class WebServiceAuthentication

[[line 18](#)]

**Provides application wide authentication using Basic HTTP Headers**

- **Package** PHPRum
- **Sub-Package** Security
- **Author** Darnell Shinbine

*bool* function WebServiceAuthentication::authenticated() [[line 25](#)]

**returns true if user authenticated**

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

*void* function WebServiceAuthentication::authenticateSecret(\$uid, \$secret) [[line 42](#)]

**Function Parameters:**

- *string* **\$uid** unique value representing user
- *string* **\$secret** secret

**authenticate secret**

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

*string* function WebServiceAuthentication::getAuthSecret() [*line 95*]

### **gets auth secret**

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

*string* function WebServiceAuthentication::getAuthUser() [*line 77*]

### **gets auth user**

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

*bool* function WebServiceAuthentication::isAuthUserSet() [*line 113*]

### **returns true if auth is set**

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

*void* function WebServiceAuthentication::setAuthUser(\$uid) [*line 61*]

#### **Function Parameters:**

- *string* **\$uid** authenticated user

### **sets auth user**

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

*void* function WebServiceAuthentication::signout() [*line* [131](#)]  
**perform sign out (does not end session)**

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

# controllertestcasebase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

`require_once __ROOT__.'system/web/webcontrols/form.class.php'` [*line [13](#)*]

**include Form, required to load constsants**

# htmltestreporter.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

`require_once __LIB_PATH__.'./simpletest/reporter.php'` [*line [14](#)*]

**include simpletest framework**

# migrationstestcasebase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# testcasebase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

`require_once __LIB_PATH__.'./simpletest/unit_tester.php'` [line [14](#)]

**include simpletest framework**

# testreporter.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

`require_once __LIB_PATH__.'./simpletest/reporter.php'` [*line [14](#)*]

**include simpletest framework**

# unittestcasebase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class ControllerTestCaseBase

[line [25](#)]

**Provides base functionality for the ControllerTestCase**

- **Package** PHPRum
- **Sub-Package** TestCase
- **Property** string \$response: response as string
- **Property** [XMLEntity](#) \$responseAsXMLEntity: response as XMLEntity
- **Author** Darnell Shinbine
- **Abstract Element**

**ControllerTestCaseBase::\$controller**

*ControllerBase* = null [line [31](#)]

**instance of the controller**

- **Access** protected

## ControllerTestCaseBase::\$view

*View* = null [*line* [37](#)]

## View object

- **Access** protected

Constructor *void* function ControllerTestCaseBase::\_\_construct(\$testCase) [*line* [58](#)]

### **Function Parameters:**

- *string* **\$testCase** Name of test case

## Constructor

- **Access** public

*bool* function ControllerTestCaseBase::assertAssigned(\$var, [\$message = ""]) [*line* [275](#)]

### **Function Parameters:**

- *string* **\$var** variable to test
- *string* **\$message** fail message

## asserts true if var is assigned

- **Access** protected

*bool* function ControllerTestCaseBase::assertMessage(\$responseMessage, [\\\$messageType = null], [\$message = "], \$messageType) [*line 316*]

**Function Parameters:**

- *string* **\$responseMessage** response message
- [\*AppMessageType\*](#) **\$messageType** message type
- *string* **\$message** fail message
- [\*AppMessageType\*](#) **\\\$messageType**

**asserts true if message exists**

- **Access** protected

*bool* function ControllerTestCaseBase::assertNoResponse(\$string, [\$message = "], \$response) [*line 261*]

**Function Parameters:**

- *string* **\$response** response
- *string* **\$message** fail message
- **\$string**

**asserts true if no response exists that matches \$string**

- **Access** protected

*bool* function ControllerTestCaseBase::assertRedirectedTo(\$forward, [\$forwardParams = array()], [\$message = "])

[line [294](#)]

**Function Parameters:**

- *string* **\$forward** forward
- *array* **\$forwardParams** forward parameters
- *string* **\$message** fail message

**asserts true if controller redirected**

- **Access** protected

*bool* function ControllerTestCaseBase::assertResponse(\$string, [\$message = "], \$response) [line [247](#)]

**Function Parameters:**

- *string* **\$response** response
- *string* **\$message** fail message
- **\$string**

**asserts true if response exists**

- **Access** protected

*void* function ControllerTestCaseBase::get([\$params = array()]) [line [152](#)]

**Function Parameters:**

- *array* **\$params** data to send

**simulate a get request**

- **Access** protected

*void* function ControllerTestCaseBase::post([\$params = array()]) [*line* [165](#)]

**Function Parameters:**

- *array* **\$params** data to send

**simulate a post request**

- **Access** protected

*string* function ControllerTestCaseBase::response() [*line* [211](#)]

**returns the response**

- **Access** protected

*XMLEntity* function ControllerTestCaseBase::responseAsXMLEntity() [*line* [232](#)]

**returns the response as XMLEntity object**

- **Access** protected

*void* function ControllerTestCaseBase::setUp() [*line* [71](#)]

**setup test module**

- **Access** public

*void* function ControllerTestCaseBase::submit(\$formId, [\$data = array()]) [*line* [178](#)]

**Function Parameters:**

- *string* **\$formId** id of the form
- *array* **\$data**

**simulate a form submission**

- **Access** protected

*void* function ControllerTestCaseBase::tearDown() [*line* [97](#)]

**clean test module**

- **Access** public

## Class HTMLTestReporter

[*line* [24](#)]

**Provides HTML reporting for test cases**



- **Package** PHPRum
- **Sub-Package** TestCase
- **Author** Darnell Shinbine

*void function* HTMLTestReporter::paintException(\$exception) [*line* [88](#)]

***Function Parameters:***

- *Exception* **\$exception** Exception to display.

**Paints a PHP exception.**

*void function* HTMLTestReporter::paintFooter(\$test\_name) [*line* [62](#)]

***Function Parameters:***

- *string* **\$test\_name** Name class of test.

**Paints the end of the test with a summary of the passes and failures.**

- **Access** public

*void function* HTMLTestReporter::paintHeader(\$test\_name) [*line* [38](#)]

***Function Parameters:***

- *string* **\$test\_name** Name class of test.

**Paints the top of the web page setting the title to the name of the starting test.**

*void function* HTMLTestReporter::setElapsedTime(\$elapsed) [*line* [103](#)]

***Function Parameters:***

- *float* **\$elapsed** time in s

**set elapsed time**

- **Access** public

## Class MigrationsTestCaseBase

*[line [17](#)]*

**Provides base functionality for the UnitTestCase**

- **Package** PHPRum
- **Sub-Package** TestCase
- **Author** Darnell Shinbine
- **Abstract Element**

**MigrationsTestCaseBase::\$state**

*array = array() [line [23](#)]*

**states**

- **Access** protected

*void function MigrationsTestCaseBase::setUp() [line [31](#)]*

**setup test module**

- **Access** public

*void* function MigrationsTestCaseBase::tearDown() [[line 41](#)]  
**clean test module**

- **Access** public

*XMLEntity* function MigrationsTestCaseBase::xxx() [[line 51](#)]  
**returns the response as XMLEntity object**

- **Access** protected

## Class TestCaseBase

[[line 26](#)]

**Provides base functionality for the TestCase**

- **Package** PHPRum
- **Sub-Package** TestCase
- **Property** string \$fixtures: comma seperated string containing fixture filenames
- **Author** Darnell Shinbine

- **Abstract Element**

**TestCaseBase::\$fixtures**

*string* = " [\[line 32\]](#)

**fixtures to load**

- **Access** protected

*void* function TestCaseBase::cleanup() [\[line 155\]](#)

**called after every test**

- **Access** protected

*array* function TestCaseBase::getFixtureVariables() [\[line 136\]](#)

**returns an associative array of fixture variables used for dynamically inserting values inside fixtures**

- **Access** protected

*bool* function TestCaseBase::loadCSVFixture(\$fixture) [\[line 293\]](#)

**Function Parameters:**

- *string* **\$fixture** name of fixture

**load csv fixture**

- **Access** protected

*void function* TestCaseBase::loadFixtures(\$fixtures) [*line* [165](#)]

***Function Parameters:***

- *array* **\$fixtures** array of fixtures

## load fixtures

- **Access** protected

*void function* TestCaseBase::loadSQLFixture(\$fixture) [*line* [321](#)]

***Function Parameters:***

- *string* **\$fixture** name of fixture

## load sql fixture

- **Access** protected

*void function* TestCaseBase::loadXMLFixture(\$fixture) [*line* [230](#)]

***Function Parameters:***

- *string* **\$fixture** name of fixture

## load xml fixture

- **Access** protected

*void* function TestCaseBase::prepare() [[line 146](#)]  
**called before every test**

- **Access** protected

*void* function TestCaseBase::run(&\$reporter, \$reporter) [[line 106](#)]  
**Function Parameters:**

- [TestReporter](#) **\$reporter** TestReporter object
- **&\$reporter**

**run report**

- **Access** public

## Class TestReporter

[[line 24](#)]

**Provides basic reporting for test cases**

- **Package** PHPRum
- **Sub-Package** TestCase
- **Author** Darnell Shinbine

*void* function TestReporter::setElapsedTime(\$elapsed) [[line 39](#)]

**Function Parameters:**

- *float* **\$elapsed** time in s

**set elapsed time**

- **Access** public

## Class UnitTestCaseBase

[[line 17](#)]

**Provides base functionality for the UnitTestCase**

- **Package** PHPRum
- **Sub-Package** TestCase
- **Author** Darnell Shinbine
- **Abstract Element**

Constructor *void* function UnitTestCaseBase::\_\_construct(\$testCase) [[line 27](#)]

**Function Parameters:**

- *string* **\$testCase** Name of test case

## Constructor

- **Access** public

*void* function UnitTestCaseBase::setUp() [*line* [37](#)]  
**setup test module**

- **Access** public

*void* function UnitTestCaseBase::tearDown() [*line* [47](#)]  
**clean test module**

- **Access** public



# timer.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class Timer

[line [18](#)]

**Provides basic timer functionality**

- **Package** PHPRum
- **Sub-Package** Utils
- **Author** Darnell Shinbine

Constructor *void* function Timer::\_\_construct([\$autostart = false]) [line [58](#)]

**Function Parameters:**

- *bool* **\$autostart** Specifies whether to automatically start the timer

### Constructor

- **Access** public

*real* function Timer::elapsed() [[line 141](#)]

**return elapsed time from start, if timer has been stopped then time will be calculated based on stop time**

- **Access public**

*real* function Timer::pause() [[line 87](#)]

**pause the timer**

- **Access public**

*void* function Timer::reset() [[line 161](#)]

**reset timer**

- **Access public**

*real* function Timer::resume() [[line 103](#)]

**resume the timer**

- **Access public**

*void* function Timer::start() [[line 70](#)]

**start timer**

- **Access** public

*void* function Timer::stop() [[line 119](#)]

**stop timer**

- **Access** public

# booleanvalidator.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# datetimevalidator.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# emailvalidator.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# enumvalidator.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# filesizevalidator.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# filetypevalidator.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# integervalidator.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# lengthvalidator.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# matchvalidator.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# numericvalidator.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# patternvalidator.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# rangevalidator.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# requiredvalidator.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# uniquevalidator.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# urlvalidator.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# validatorbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# validatorcollection.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class BooleanValidator

[line [20](#)]

**Provides basic validation for web controls**

- **Package** PHPRum
- **Sub-Package** Validators
- **Property** string \$controlId: Control Id
- **Author** Darnell Shinbine

*void* function BooleanValidator::onLoad() [line [27](#)]  
**on load**

- **Access** protected

*void* function BooleanValidator::validate() [line [41](#)]

**sets the controlId and prepares the control attributes**

- **Access** public

## Class DateTimeValidator

[*line* [20](#)]

**Provides basic validation for web controls**

- **Package** PHPRum
- **Sub-Package** Validators
- **Property** string \$controlId: Control Id
- **Author** Darnell Shinbine

*void* function DateTimeValidator::onLoad() [*line* [27](#)]  
**on load**

- **Access** protected

*void* function DateTimeValidator::validate() [*line* [41](#)]  
**sets the controlId and prepares the control attributes**

- **Access** public

# Class EmailValidator

[[line 19](#)]

**Provides basic validation for web controls**

- **Package** PHPRum
- **Sub-Package** Validators
- **Property** string \$controlId: Control Id
- **Author** Darnell Shinbine

Constructor *void* function EmailValidator::\_\_construct([\$errorMessage = ""]) [[line 27](#)]

**Function Parameters:**

- *string* **\$errorMessage** error message

## EmailAddressValidator

- **Access** public

*void* function EmailValidator::onLoad() [[line 38](#)]

**on load**

- **Access** protected

# Class EnumValidator

[line [17](#)]

**Provides basic validation for web controls**

- **Package** PHPRum
- **Sub-Package** Validators
- **Author** Darnell Shinbine

Constructor *void* function EnumValidator::\_\_construct(\$values, [\$errorMessage = ""]) [line [31](#)]

**Function Parameters:**

- *string* **\$errorMessage** error message
- *array* **\$values**

## URLValidator

- **Access** public

*void* function EnumValidator::onLoad() [line [43](#)]

**on load**

- **Access** protected

*void* function EnumValidator::validate() [*line* [65](#)]

**sets the controlId and prepares the control attributes**

- **Access** public

## Class FileSizeValidator

[*line* [20](#)]

**Provides basic validation for web controls**

- **Package** PHPRum
- **Sub-Package** Validators
- **Property** string \$controlId: Control Id
- **Author** Darnell Shinbine

Constructor *void* function FileSizeValidator::\_\_construct(\$maxSize, [\$errorMessage = ""]) [*line* [36](#)]

**Function Parameters:**

- *double* **\$maxSize** max size
- *string* **\$errorMessage** error message

### FileSizeValidator



- **Access** public

*void* function FileSizeValidator::onLoad() [[line 49](#)]  
**on load**

- **Access** protected

*bool* function FileSizeValidator::validate() [[line 63](#)]  
**validates the control**

- **Access** public

## Class FileTypeValidator

[[line 20](#)]

**Provides basic validation for web controls**

- **Package** PHPRum
- **Sub-Package** Validators
- **Property** string \$controlId: Control Id
- **Author** Darnell Shinbine

Constructor *void* function FileTypeValidator::\_\_construct([\$types = array()], [\$errorMessage = ""]) [[line 36](#)]

### ***Function Parameters:***

- *array* **\$types** array of types
- *string* **\$errorMessage** error message

## **FileTypeValidator**

- **Access** public

*void* function FileTypeValidator::addType(\$type) [[line 50](#)]

### ***Function Parameters:***

- *string* **\$type** file type

**adds a file type to the list of allowed types**

- **Access** public

*void* function FileTypeValidator::onLoad() [[line 61](#)]

**on load**

- **Access** protected

*bool* function FileTypeValidator::validate() [[line 75](#)]

**validates the control**

- **Access** public

## Class IntegerValidator

[[line 20](#)]

**Provides basic validation for web controls**

- **Package** PHPRum
- **Sub-Package** Validators
- **Property** string \$controlId: Control Id
- **Author** Darnell Shinbine

*void* function IntegerValidator::onLoad() [[line 27](#)]  
**on load**

- **Access** protected

*void* function IntegerValidator::validate() [[line 41](#)]  
**sets the controlId and prepares the control attributes**

- **Access** public

# Class LengthValidator

[line [20](#)]

Provides basic validation for web controls

- **Package** PHPRum
- **Sub-Package** Validators
- **Property** string \$controlId: Control Id
- **Author** Darnell Shinbine

Constructor *void* function LengthValidator::\_\_construct(\$min, \$max, [\$errorMessage = ""]) [line [43](#)]

## **Function Parameters:**

- *double* **\$min** min value
- *double* **\$max** max value
- *string* **\$errorMessage** error message

## LengthValidator

- **Access** public

*void* function LengthValidator::onLoad() [line [57](#)]  
**on load**

- **Access** protected

*void* function LengthValidator::validate() [*line* [71](#)]

**sets the controlId and prepares the control attributes**

- **Access** public

## Class MatchValidator

[*line* [20](#)]

**Provides basic validation for web controls**

- **Package** PHPRum
- **Sub-Package** Validators
- **Property** string \$controlId: Control Id
- **Author** Darnell Shinbine

**MatchValidator::\$controlToMatch**

*InputBase* = [*line* [26](#)]

**control to match**

- **Access** protected

Constructor *void* function MatchValidator::\_\_construct(\\&\$controlToMatch, [\$errorMessage = "], \$controlToMatch) [*line 36*]

**Function Parameters:**

- [InputBase](#) **\$controlToMatch** control to match
- *string* **\$errorMessage** error message
- [InputBase](#) **\\&\$controlToMatch**

## MatchValidator

- **Access** public

*void* function MatchValidator::onLoad() [*line 49*]  
**on load**

- **Access** protected

*void* function MatchValidator::validate() [*line 63*]  
**sets the controlId and prepares the control attributes**

- **Access** public

# Class NumericValidator

[line 20]

Provides basic validation for web controls

- **Package** PHPRum
- **Sub-Package** Validators
- **Property** string \$controlId: Control Id
- **Author** Darnell Shinbine

*void* function NumericValidator::onLoad() [line 27]

**on load**

- **Access** protected

*void* function NumericValidator::validate() [line 41]

**sets the controlId and prepares the control attributes**

- **Access** public

# Class PatternValidator

[line 20]

Provides basic validation for web controls

- **Package** PHPRum
- **Sub-Package** Validators
- **Property** string \$controlId: Control Id
- **Author** Darnell Shinbine

Constructor *void* function PatternValidator::\_\_construct(\$pattern, [\$errorMessage = "]) [[line 36](#)]

**Function Parameters:**

- *string* **\$pattern** pattern
- *string* **\$errorMessage** error message

## PatternValidator

- **Access** public

*void* function PatternValidator::onLoad() [[line 49](#)]

**on load**

- **Access** protected

*void* function PatternValidator::validate() [[line 63](#)]

**sets the controlId and prepares the control attributes**

- **Access** public



# Class RangeValidator

[*line* [20](#)]

**Provides basic validation for web controls**

- **Package** PHPRum
- **Sub-Package** Validators
- **Property** string \$controlId: Control Id
- **Author** Darnell Shinbine

Constructor *void* function RangeValidator::\_\_construct(\$min, \$max, [\$errorMessage = "]) [*line* [43](#)]

## **Function Parameters:**

- *double* **\$min** min value
- *double* **\$max** max value
- *string* **\$errorMessage** error message

## **RangeValidator**

- **Access** public

*void* function RangeValidator::onLoad() [*line* [57](#)]  
**on load**

- **Access** protected

*void* function RangeValidator::validate() [*line* [71](#)]

**sets the controlId and prepares the control attributes**

- **Access** public

## Class RequiredValidator

[*line* [20](#)]

**Provides basic validation for web controls**

- **Package** PHPRum
- **Sub-Package** Validators
- **Property** string \$controlId: Control Id
- **Author** Darnell Shinbine

*void* function RequiredValidator::onLoad() [*line* [27](#)]

**on load**

- **Access** protected

*bool* function RequiredValidator::validate() [*line* [42](#)]

**validates the control**

- **Access** public

## Class UniqueValidator

[[line 20](#)]

**Provides basic validation for web controls**

- **Package** PHPRum
- **Sub-Package** Validators
- **Property** string \$controlId: Control Id
- **Author** Darnell Shinbine

*void* function UniqueValidator::onLoad() [[line 27](#)]  
**on load**

- **Access** protected

*void* function UniqueValidator::validate() [[line 41](#)]  
**sets the controlId and prepares the control attributes**

- **Access** public

# Class URLValidator

[*line* [19](#)]

**Provides basic validation for web controls**

- **Package** PHPRum
- **Sub-Package** Validators
- **Property** string \$controlId: Control Id
- **Author** Darnell Shinbine

Constructor *void* function URLValidator::\_\_construct([\$errorMessage = ""]) [*line* [27](#)]

## **Function Parameters:**

- *string* **\$errorMessage** error message

## **URLValidator**

- **Access** public

*void* function URLValidator::onLoad() [*line* [38](#)]

## **on load**

- **Access** protected

# Class ValidatorBase

[line [20](#)]

Provides basic validation for web controls

- **Package** PHPRum
- **Sub-Package** Validators
- **Property** string \$errorMessage: error message
- **Author** Darnell Shinbine
- **Abstract Element**

**ValidatorBase::\$controlToValidate**

*InputBase* = [line [33](#)]

**control to validate**

- **Access** protected

**ValidatorBase::\$errorMessage**

*string* = [line [26](#)]

**error message**

- **Access** protected

Constructor *void* function ValidatorBase::\_\_construct([\$errorMessage = ""]) [[line 42](#)]

**Function Parameters:**

- *string* **\$errorMessage** error message

## ValidatorBase

- **Access** public

*void* function ValidatorBase::onLoad() [[line 127](#)]

**on load**

- **Access** protected

*void* function ValidatorBase::setControlToValidate(\\&\$controlToValidate, \$controlToValidate) [[line 105](#)]

**Function Parameters:**

- [InputBase](#) **\$controlToValidate** control to validate
- [InputBase](#) **\\&\$controlToValidate**

## set control to validate

- **Access** public

*void* function ValidatorBase::setErrorMessage(\$errorMessage) [*line* [93](#)]

**Function Parameters:**

- *string* **\$errorMessage** error message

**set error message**

- **Access** public

*bool* function ValidatorBase::validate(\$controlToValidate) [*line* [119](#)]

**Function Parameters:**

- [InputBase](#) **\$controlToValidate** control to validate

**validates the control**

- **Abstract Element**
- **Access** public

## Class ValidatorCollection

[*line* [19](#)]

**Represents a Collection of Validator objects**

- **Package** PHPRum
- **Sub-Package** Validators
- **Author** Darnell Shinbine

Constructor *void* function ValidatorCollection::\_\_construct(\\&\$control, \$parent) [*line 35*]

**Function Parameters:**

- [InputBase](#) \$parent instance of a InputBase object
- [InputBase](#) \\&\$control

## Constructor

- **Access** public

*void* function ValidatorCollection::add(\$item) [*line 72*]

**Function Parameters:**

- *Validator* \$item

## add Validator to Collection

- **Access** public

*bool* function ValidatorCollection::contains(\$item) [*line 92*]

**Function Parameters:**

- *mixed* \$item item



**returns true if item array contains item**

- **Access** public

# controllerbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# formajaxpostevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# formajaxposteventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# formpostevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# formposteventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# gridviewcolumnajaxpostevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# gridviewcolumnajaxposteventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# gridviewcolumnpostevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# gridviewcolumnposteventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# inputajaxchangeevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# inputajaxchangeeventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# inputajaxpostevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# inputajaxposteventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# inputchangeevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# inputchangeeventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# inputpostevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# inputposteventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# loginformsubmitevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# loginformsubmiteventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# masterviewinitevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# masterviewiniteventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# masterviewloadevent.class.php

- **Package** PHPRum
- **Author** Darnell Shimbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# masterviewloadeventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# pagepostevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# pageposteventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# pagerequestevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# pagerequesteventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# reportdetailsevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# reportdetailseventhander.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# reportfooterevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# reportfootereventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# reportgroupfooterevent.class.php

- **Package** PHPRum
- **Author** Darnell Shimbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# reportgroupfootereventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# reportgroupheaderevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# reportgroupheadereventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# reportheaderevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# reportheadereventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# webcontrolcreateevent.class.php

- **Package** PHPRum
- **Author** Darnell Shimbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# webcontrolcreateeventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# webcontrolinitevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# webcontroliniteventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# webcontrolloadevent.class.php

- **Package** PHPRum
- **Author** Darnell Shimbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# webcontrolloadeventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# webcontrolprerenderevent.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# webcontrolprerendereventhandler.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# formmodelbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# forwardmethodtype.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# pagecontrollerbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# jsonservicebase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# restwebservicebase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# soapwebservicebase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# webservicebase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# webserviceMethod.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# button.class.php

- **Package** PHPRum
- **Filesource** [Source Code for this file](#)
- **License** see

# checkbox.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# checkboxlist.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# dateselector.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# datetimeselector.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# dropdownlist.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# fieldset.class.php

- **Package** PHPRum
- **Filesource** [Source Code for this file](#)
- **License** see

# filebrowser.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# form.class.php

- **Package** PHPRum
- **Filesource** [Source Code for this file](#)
- **License** see

# gridview.class.php

- **Package** PHPRum
- **Filesource** [Source Code for this file](#)
- **License** see



# gridviewbutton.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# gridviewcheckbox.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# gridviewcolumn.class.php

- **Package** PHPRum
- **Filesource** [Source Code for this file](#)
- **License** see

# gridviewcolumncollection.class.php

- **Package** PHPRum
- **Filesource** [Source Code for this file](#)
- **License** see

# gridviewcontrolbase.class.php

- **Package** PHPRum
- **Filesource** [Source Code for this file](#)
- **License** see

# gridviewdropdownmenu.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# gridviewimage.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# gridviewlink.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# gridviewtextbox.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# hyperlink.class.php

- **Package** PHPRum
- **Filesource** [Source Code for this file](#)
- **License** see

# inputbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# label.class.php

- **Package** PHPRum
- **Filesource** [Source Code for this file](#)
- **License** see

# listbox.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# listcontrolbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# listitemcollection.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# loginform.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# masterview.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# page.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# radiobutton.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# radiobuttonlist.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# radiogroup.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# reportgroup.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# reportview.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# textarea.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# textbox.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# timeselector.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# treenode.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# treenodecollection.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# treeview.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# view.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# webcontrolattributecollection.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# webcontrolbase.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see



# webcontrolcollection.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class Button

[*line* [23](#)]

**Represents a Button Control**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$text: Specifies button text
- **Property** string \$src: Specifies button image source
- **Property** string \$submitText: Specifies button text to display on submit
- **Property** bool \$disableOnSubmit: Specifies whether to disable all other buttons on submit
- **Author** Darnell Shinbine

**Button::\$disableOnSubmit**

*bool* = false [*line* [47](#)]

**specifies whether to disable all other buttons on submit**

- **Access** protected

#### **Button::\$src**

*string* = " [[line 35](#)]

**specifies button image source**

- **Access** protected

#### **Button::\$submitText**

*string* = " [[line 41](#)]

**specifies button text to display on submit**

- **Access** protected

#### **Button::\$text**

*string* = " [[line 29](#)]

**specifies button text**

- **Access** protected

Constructor *void* function Button::\_\_construct(\$controlId, [\$text = "]) [[line 57](#)]

***Function Parameters:***

- *string* **\$controlId** Control Id
- *string* **\$text** Button text

## Constructor

- **Access** public

*DomObject* function Button::getDomObject() [*line* [145](#)]

### **getDomObject**

returns a DomObject representing control

- **Access** public

*bool* function Button::onLoad() [*line* [192](#)]

### **onLoad**

called when control is loaded

- **Access** protected

*void* function Button::onRequest(&\$request) [*line* [219](#)]

#### **Function Parameters:**

- *array* **&\$request** request data

### **onRequest**

process the HTTP request array

- **Access** protected

## Class CheckBox

[[line 20](#)]

**Represents a CheckBox Control**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** bool \$checked: Specifies whether button is checked
- **Author** Darnell Shinbine

*DomObject* function `CheckBox::getDomObject()` [[line 68](#)]  
**returns a DomObject representing control**

- **Access** public

*bool* function `CheckBox::onLoad()` [[line 119](#)]  
**called when control is loaded**

- **Access** protected

*void* function CheckBox::onRequest(&\$request) [*line* [135](#)]

**Function Parameters:**

- *array* &\$request

**process the HTTP request array**

- **Access** protected

## Class CheckBoxList

[*line* [18](#)]

**Represents a CheckBoxList Control**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

**CheckBoxList::\$multiple**

*bool* = true [*line* [24](#)]

**size of listbox**

- **Access** protected

*DomObject* function `CheckBoxList::getDomObject()` [[line 32](#)]  
**returns a DomObject representing control**

- **Access** public

## Class ControllerBase

[[line 25](#)]

**This class handles all requests for a specific uri.**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$controllerId: Specifies the controller id
- **Property** int \$outputCache: Specifies how long to cache page output in seconds, 0 disables caching
- **Property** bool \$requireSSL: Specifies whether SSL is required for this page
- **Property** array \$allowRoles: Specifies the roles that have been granted access this page
- **Property** array \$denyRoles: Specifies the roles that have been denied access this page
- **Property** [EventCollection](#) \$events: event collection
- **Author** Darnell Shinbine
- **Abstract Element**

**ControllerBase::\$allowRoles**

*array* = array() [*line* [49](#)]

**Specifies the roles that have been granted access this page**

- **Access** protected

**ControllerBase::\$controllerId**

*string* = " [*line* [31](#)]

**Specifies the controller id**

- **Access** protected

**ControllerBase::\$denyRoles**

*array* = array() [*line* [55](#)]

**Specifies the roles that have been denied access this page**

- **Access** protected

**ControllerBase::\$events**

*EventCollection* = null [*line* [61](#)]

**event collection**

- **Access** protected

## ControllerBase::\$outputCache

*int* = 0 [[line 37](#)]

**Specifies how long to cache page output in seconds, 0 disables caching**

- **Access** protected

## ControllerBase::\$requireSSL

*bool* = false [[line 43](#)]

**Specifies whether the controller requires an SSL connection**

- **Access** protected

Constructor *void* function ControllerBase::\_\_construct(\$controllerId) [[line 70](#)]

### **Function Parameters:**

- *string* **\$controllerId** Controller Id

## Constructor

- **Access** public

*string* function ControllerBase::getCacheId(&\$request) [[line 154](#)]

### **Function Parameters:**

- [HttpRequest](#) **&\$request** HttpRequest object



**this method will generate and return a cache id**

- **Access** public

*array function ControllerBase::getRoles() [line [142](#)]*

**this method return an array of roles that have been granted access**

- **Access** public

*View function ControllerBase::getView(\\&\$request, &\$request) [line [166](#)]*

**Function Parameters:**

- [\*HttpRequest\*](#) **&\$request** HttpRequest object
- [\*HttpRequest\*](#) **\\&\$request**

**return view component for rendering**

- **Abstract Element**
- **Access** public

*void function ControllerBase::onLoad() [line [131](#)]*

**Event called when object created**

- **Access** protected

## Class DateSelector

*[line 24]*

**handles date control element creation (string) abstracts away the presentation logic and data access layer the server-side control for WebWidgets**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** int \$yearMin: year minimum
- **Property** int \$yearMax: year maximum
- **Property** bool \$allowNull: specifies whether to allow null values
- **Author** Darnell Shinbine

### DateSelector::\$allowNull

*mixed = false [line 42]*

**specifies whether to allow nulls**

- **Access** protected

### DateSelector::\$yearMax

*mixed = 2020 [line 36]*

**specifies the year range**

- **Access** protected

**DateSelector::\$yearMin**

*mixed* = 1900 [*line* [30](#)]

**specifies the year range**

- **Access** protected

Constructor *void* function DateSelector::\_\_construct(\$controlId, [\$default = null]) [*line* [51](#)]

**Function Parameters:**

- **\$controlId**
- **\$default**

**Constructor**

- **Access** public

*void* function DateSelector::getDomObject(0) [*line* [185](#)]

**Function Parameters:**

- *none* **0**

**returns widget object**

- **Access** public

*void* function DateSelector::onLoad() [*line* [170](#)]  
**called when control is loaded**

- **Access** protected

*void* function DateSelector::onRequest(&\$httpRequest) [*line* [116](#)]  
**Function Parameters:**

- *array* &\$httpRequest

**process the HTTP request array**

- **Access** protected

## Class DateTimeSelector

[*line* [24](#)]

**handles date control element creation (string) abstracts away the presentation logic  
 and data access layer the server-side control for WebWidgets**

- **Package** PHPRum

- **Sub-Package** Web
- **Property** int \$yearMin: year minimum
- **Property** int \$yearMax: year maximum
- **Property** bool \$allowNull: specifies whether to allow null values
- **Author** Darnell Shinbine

#### **DateTimeSelector::\$allowNull**

*mixed* = false [[line 42](#)]

**specifies whether to allow nulls**

- **Access** protected

#### **DateTimeSelector::\$yearMax**

*mixed* = 2020 [[line 36](#)]

**specifies the year range**

- **Access** protected

#### **DateTimeSelector::\$yearMin**

*mixed* = 1900 [[line 30](#)]

**specifies the year range**

- **Access** protected

Constructor *void* function DateTimeSelector::\_\_construct(\$controlId, [\$default = null]) [[line 51](#)]

**Function Parameters:**

- **\$controlId**
- **\$default**

## Constructor

- **Access** public

*void* function DateTimeSelector::getDomObject(0) [[line 199](#)]

**Function Parameters:**

- *none* **0**

## returns widget object

- **Access** public

*void* function DateTimeSelector::onLoad() [[line 115](#)]

## called when control is loaded

- **Access** protected

*void* function DateTimeSelector::onRequest(&\$httpRequest) [[line 129](#)]

**Function Parameters:**

- **array &\$httpRequest**

**process the HTTP request array**

- **Access** protected

## Class DropDownList [line [18](#)]

**Represents a DropDownList Control**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

**DropDownList::\$listSize**

*int* = 1 [line [24](#)]

**size of listbox**

- **Access** protected

**DropDownList::\$multiple**

*bool* = false [[line 30](#)]

## size of listbox

- **Access** protected

# Class Fieldset

[[line 22](#)]

## Represents a Fieldset

- **Package** PHPRum
- **Sub-Package** Web
- **Property** bool \$ajaxPostBack: specifies whether to perform ajax postback, Default is false
- **Property** bool \$ajaxValidation: specifies whether to perform ajax validation, Default is false
- **Property** string \$legend: fieldset legend
- **Author** Darnell Shinbine

## Fieldset::\$legend

*string* = " [[line 28](#)]

## Fieldset legend

- **Access** protected



Constructor *void* function Fieldset::\_\_construct(\$controlId) [*line 38*]

**Function Parameters:**

- *string* **\$controlId** Control Id

## Constructor

- **Access** public

*void* function Fieldset::add(\$control, &\$control) [*line 101*]

**Function Parameters:**

- [InputBase](#) **&\$control** instance of an InputBase
- [InputBase](#) **\$control**

## adds child control to collection

- **Access** public

*void* function Fieldset::fillDataSource(&\$ds, \$ds) [*line 160*]

**Function Parameters:**

- *ArrayAccess* **\$ds** data source to fill
- *ArrayAccess* **&\$ds**

## update data source with Control data

- **Access** public

*bool* function Fieldset::focus() [*line* [123](#)]

**sets focus to the control**

- **Access** public

*DomObject* function Fieldset::getDomObject() [*line* [189](#)]

**returns a DomObject representing control**

- **Access** public

*bool* function Fieldset::onLoad() [*line* [112](#)]

**called when control is loaded**

- **Access** protected

*void* function Fieldset::readDataSource(&\$ds, \$ds) [*line* [175](#)]

**Function Parameters:**

- *ArrayAccess* **\$ds** data source to read
- *ArrayAccess* **&\$ds**

**update Control with data from the data source**

- **Access** public

*bool* function Fieldset::validate([&\$errMsg = ''], [&\$controlToFocus = null], \$errMsg) [*line* [139](#)]

**Function Parameters:**

- *string* **\$errMsg** error message
- **&\$errMsg**
- [InputBase](#) **&\$controlToFocus**

**validate all controls in form object**

- **Access** public

## Class FileBrowser

[*line* [18](#)]

**Represents a FileBrowser Control**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function FileBrowser::\_\_construct(\$controlId) [*line* [26](#)]

### ***Function Parameters:***

- *string* **\$controlId** Control Id

### **Constructor**

- **Access** public

*DomObject* function *FileBrowser::getDomObject()* [[line 125](#)]  
**returns a DomObject representing control**

- **Access** public

*array* function *FileBrowser::getFileInfo()* [[line 39](#)]  
**returns information on the uploaded file**

- **Access** public

*string* function *FileBrowser::getFileRawData()* [[line 57](#)]  
**return raw file data**

- **Access** public

*bool* function *FileBrowser::onLoad()* [[line 140](#)]  
**called when control is loaded**

- **Access** protected

*void* function FileBrowser::onRequest(&\$request) [[line 157](#)]

**Function Parameters:**

- *array* &\$request

**process the HTTP request array**

- **Access** protected

## Class Form

[[line 35](#)]

**Represents a Form**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$action: form action
- **Property** string \$method: form submit method
- **Property** string \$encodeType: form encoding type
- **Property** string \$forward: controller to forward to
- **Property** bool \$ajaxPostBack: specifies whether to perform ajax postback, Default is false

- **Property** bool \$ajaxValidation: specifies whether to perform ajax validation, Default is false
- **Property** bool \$autoFocus: specifies whether to auto focus
- **Property** bool \$hiddenField: specifies whether to check for hidden honeypot field before processing request
- **Property** string \$submitted: specifies if form was submitted
- **Property** RequestParameterCollection \$parameters: form parameters
- **Author** Darnell Shinbine

#### Form::\$action

*string* = " [line 48]

**URL to send form data, Default is self**

- **Access** protected

#### Form::\$ajaxPostBack

*bool* = false [line 78]

**turn on or off ajax post backs**

- **Access** protected

#### Form::\$ajaxValidation

*bool* = false [line 84]

**turn on or off ajax validation**

- **Access** protected

#### **Form::\$autoFocus**

*bool* = false [[line 72](#)]

**turn on or off auto focusing**

- **Access** protected

#### **Form::\$encodeType**

*string* = 'application/x-www-form-urlencoded' [[line 60](#)]

**Form encoding type**

- **Access** protected

#### **Form::\$forward**

*string* = " [[line 66](#)]

**controller to forward to, Default is the current controller**

- **Access** protected

#### **Form::\$hiddenField**

*bool* = false [[line 90](#)]

**specifies whether to check for hidden honeypot field before processing request**

- **Access** protected

#### **Form::\$method**

*string* = 'POST' [[line 54](#)]

**method to send form data, Default is 'POST'**

- **Access** protected

#### **Form::\$parameters**

*StringDictionary* = null [[line 102](#)]

**array of variables to post on submit**

- **Access** protected

#### **Form::\$submitted**

*bool* = false [[line 96](#)]

**set if the form was submitted**

- **Access** protected

Constructor *void* function Form::\_\_construct(\$controlId) [[line 118](#)]

***Function Parameters:***



- *string* **\$controlId** Control Id

## Constructor

- **Access** public

*void* function Form::add(\$control, &\$control) [*line* [278](#)]

### **Function Parameters:**

- [InputBase](#) &**\$control** instance of an InputBase
- [WebControlBase](#) **\$control**

## adds child control to collection

- **Access** public

*void* function Form::addParameter(\$key, \$value) [*line* [298](#)]

### **Function Parameters:**

- *string* **\$key** key
- *string* **\$value** value

## add parameter

- **Access** public

*void* function Form::begin([\$args = array()]) [[line 387](#)]

**Function Parameters:**

- *array* **\$args** attribute parameters

**renders form open tag**

- **Access** public

*void* function Form::end() [[line 399](#)]

**renders form close tag**

- **Access** public

*DomObject* function Form::getDomObject() [[line 410](#)]

**returns a DomObject representing control**

- **Access** public

*void* function Form::getFormDomObject() [[line 493](#)]

**returns form as widget**

- **Access** protected

*void* function Form::onDataBind() [[line 691](#)]

**bind all child controls in form to datasource**

- **Access** protected

*bool* function Form::onLoad() [[line 584](#)]

**called when control is loaded**

- **Access** protected

*void* function Form::onPost(&\$request) [[line 670](#)]

**Function Parameters:**

- *array* **&\$request** request data

**handle post events**

- **Access** protected

*void* function Form::onRequest(&\$request) [[line 610](#)]

**Function Parameters:**

- *array* **&\$request** request array

**process the HTTP request array**

- **Access** protected

*void* function Form::save() [[line 309](#)]

**update DataSet using values from child controls**

- **Access** public

*bool* function Form::validate([&\$errMsg = "], [&\$controlToFocus = null], \$errMsg) [[line 348](#)]

**Function Parameters:**

- *string* **\$errMsg** error message
- **&\$errMsg**
- [InputBase](#) **&\$controlToFocus**

**validate all controls in form object**

- **Access** public

## Class FormAjaxPostEvent

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function FormAjaxPostEvent::\_\_construct() [*line 26*]

### Constructor

- **Access** public

## Class FormAjaxPostEventHandler

[*line 19*]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function FormAjaxPostEventHandler::\_\_construct(\$callback) [*line 27*]

### **Function Parameters:**

- *string* **\$callback** call back

### Constructor

- **Access** public

## Class FormModelBase

[line [23](#)]

**This class represents a form.**

The FormModelBase exposes 2 protected properties

- **Package** PHPRum
- **Sub-Package** Web
- **Property** array \$fields: Contains an associative array of field names mapped to field types
- **Property** array \$rules: Contains an associative array of field names mapped to rules
- **Author** Darnell Shinbine
- **Abstract Element**

### FormModelBase::\$fields

*array = array()* [line [29](#)]

**Contains an associative array of field names mapped to field types**

- **Access** protected

## FormModelBase::\$field\_mappings

```
array = array(  
    'binary' => 'System\Web\WebControls\FileBrowser',  
    'blob' => 'System\Web\WebControls\TextArea',  
    'boolean' => 'System\Web\WebControls\CheckBox',  
    'date' => 'System\Web\WebControls\DateSelector',  
    'datetime' => 'System\Web\WebControls\DateTimeSelector',  
    'enum' => 'System\Web\WebControls\DropDownList',  
    'integer' => 'System\Web\WebControls\TextBox',  
    'numeric' => 'System\Web\WebControls\TextBox',  
    'real' => 'System\Web\WebControls\TextBox',  
    'ref' => 'System\Web\WebControls\DropDownList',  
    'string' => 'System\Web\WebControls\TextBox',  
    'time' => 'System\Web\WebControls\TimeSelector'  
) [line 41]
```

## Contains field mappings

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

## FormModelBase::\$rules

```
array = array() [line 35]
```

## Contains an associative array of field names mapped to rules

- **Access** protected

## FormModelBase::\$rule\_mappings

```
array = array(  
    'boolean' => 'System\Validators\BooleanValidator',  
    'datetime' => 'System\Validators\DateTimeValidator',  
    'email' => 'System\Validators\EmailValidator',  
    'enum' => 'System\Validators\EnumValidator',  
    'filesize' => 'System\Validators\FileSizeValidator',  
    'filetype' => 'System\Validators\FileTypeValidator',  
    'integer' => 'System\Validators\IntegerValidator',  
    'length' => 'System\Validators\LengthValidator',  
    'match' => 'System\Validators\MatchValidator',  

```

```
'numeric' => 'System\Validators\NumericValidator',  
'pattern' => 'System\Validators\PatternValidator',  
'range' => 'System\Validators\RangeValidator',  
'required' => 'System\Validators\RequiredValidator',  
'unique' => 'System\Validators\UniqueValidator',  
'url' => 'System\Validators\URLValidator'  
) [line 60]
```

## Contains rule mappings

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

Constructor *void* function FormModelBase::\_\_construct() [[line 92](#)]

### Constructor

Read values from request, clean up variables, and merge get and post requests.

- **Access** protected

*Form* function FormModelBase::form(\$controllId) [[line 240](#)]

### Function Parameters:

- *string* **\$controllId** form id

## static method to return a Form object

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



`void function FormModelBase::registerFieldType($field, $type) [line 342]`

**Function Parameters:**

- *string* **\$field** field type
- *string* **\$type** path to control

## register new field type

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

`void function FormModelBase::registerRuleType($rule, $type) [line 355]`

**Function Parameters:**

- *string* **\$rule** name of rule
- *string* **\$type** path to control

## register new rule type

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

`void function FormModelBase::save() [line 231]`

## save model state

- **Abstract Element**

- **Access** public

array function FormModelBase::toArray() [[line 220](#)]  
**converts the form model into an array**

- **Access** public

## Class FormPostEvent

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function FormPostEvent::\_\_construct() [[line 26](#)]  
**Constructor**

- **Access** public

# Class FormPostEventHandler

[[line 19](#)]

Provides event handling

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function FormPostEventHandler::\_\_construct(\$callback) [[line 27](#)]

**Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class ForwardMethodType

[[line 18](#)]

Represents an application forward method

- **Package** PHPRum
- **Sub-Package** Web

- **Author** Darnell Shimbine

*ForwardMethodType* function *ForwardMethodType::Request()* [[line 37](#)]  
**Specifies the application forward method as Request**

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

*ForwardMethodType* function *ForwardMethodType::URI()* [[line 31](#)]  
**Specifies the application forward method as URI**

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

## Class GridView

[[line 49](#)]

**Represents a GridView Control**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$caption:
- **Property** int \$pageSize:
- **Property** int \$page:

- **Property** int \$filterGroupSize:
- **Property** bool \$canSort:
- **Property** bool \$canChangeOrder:
- **Property** bool \$showFilters:
- **Property** bool \$showHeader:
- **Property** bool \$showFooter:
- **Property** bool \$showPageNumber:
- **Property** bool \$showPrimaryKey:
- **Property** bool \$showList:
- **Property** bool \$autoGenerateColumns:
- **Property** string \$valueField:
- **Property** string \$orderByField:
- **Property** [GridViewColumnCollection](#) \$columns:
- **Property** bool \$multiple:
- **Property** array \$selected:
- **Property** string \$listName:
- **Property** string \$sortBy:
- **Property** string \$sortOrder:
- **Property** string \$moveCursor:
- **Property** string \$moveOrder:
- **Property** array \$filters:
- **Property** string \$onmouseover:
- **Property** string \$onmouseout:
- **Property** string \$onclick:
- **Property** string \$ondblclick:
- **Property** bool \$ajaxPostBack: specifies whether to perform ajax postback on change, Default is false
- **Author** Darnell Shinbine
- **Version** 2.0

### **GridView::\$actions**

*array* = array() [*line* [199](#)]

#### **Array of actions**

- **Access** protected

### **GridView::\$ajaxPostBack**

*bool* = false [*line* [235](#)]

**Specifies whether form will submit ajax postback on change, Default is false**

- **Access** protected

### **GridView::\$autoGenerateColumns**

*bool* = false [*line* [127](#)]

**Specifies whether to generate columns from datasource, default is false**

- **Access** protected

### **GridView::\$canChangeOrder**

*bool* = false [*line* [85](#)]

**Specifies if table order can be changed, Default is false**

- **Access** protected

#### **GridView::\$canSort**

*bool* = true [[line 79](#)]

**Specifies if table is sortable, Default is true**

- **Access** protected

#### **GridView::\$caption**

*string* = " [[line 55](#)]

**Table caption, Default is none**

- **Access** protected

#### **GridView::\$columns**

*GridViewColumnCollection* = [[line 205](#)]

**collection of columns**

- **Access** protected

#### **GridView::\$filterGroupSize**

*int* = 10 [[line 73](#)]

**Number of records to display in dropdown filter, Default is 10**

- **Access** protected

#### **GridView::\$filters**

*array = array()* [[line 187](#)]

#### **current grid filters**

- **Access** protected

#### **GridView::\$filterValues**

*array = array()* [[line 193](#)]

#### **grid filter values**

- **Access** protected

#### **GridView::\$listName**

*string = '&nbsp;'* [[line 157](#)]

#### **optional name for the list**

- **Access** protected

#### **GridView::\$moveCursor**



*string* = 0 [[line 175](#)]

**current grid moveCursor field**

- **Access** protected

**GridView::\$moveOrder**

*bool* = " [[line 181](#)]

**current grid moveCursor order**

- **Access** protected

**GridView::\$multiple**

*bool* = false [[line 133](#)]

**specifies if list can have multiple values (used when showList is true), default is false**

- **Access** protected

**GridView::\$onclick**

*string* = " [[line 223](#)]

**specifies the action to take on click events**

- **Access** protected

#### **GridView::\$ondblclick**

*string* = " [[line 229](#)]

**specifies the action to take on double click events**

- **Access** protected

#### **GridView::\$onmouseout**

*string* = " [[line 217](#)]

**specifies the action to take on onmouseout events**

- **Access** protected

#### **GridView::\$onmouseover**

*string* = " [[line 211](#)]

**specifies the action to take on mouseover events**

- **Access** protected

#### **GridView::\$orderByField**

*string* = " [[line 151](#)]

**specifies the order by field**

- **Access** protected

#### **GridView::\$page**

*int* = 1 [[line 67](#)]

**current grid page**

- **Access** protected

#### **GridView::\$pageSize**

*int* = 20 [[line 61](#)]

**Number of rows to display per page, Default is 20**

- **Access** protected

#### **GridView::\$selected**

*array* = array() [[line 139](#)]

**array of values for selected items (used when showList is true)**

- **Access** protected

#### **GridView::\$showFilters**

*bool* = false [[line 91](#)]

**Set to display column filters, Default is false**

- **Access** protected

#### **GridView::\$showFooter**

*bool* = false [[line 103](#)]

**Set to display table footer, Default is false**

- **Access** protected

#### **GridView::\$showHeader**

*bool* = true [[line 97](#)]

**Set to display column headers, Default is true**

- **Access** protected

#### **GridView::\$showList**

*bool* = false [[line 121](#)]

**Show list view, default is false**

- **Access** protected

#### **GridView::\$showPageNumber**

*bool* = true [[line 109](#)]

**Set to display table footer, Default is true**

- **Access** protected

**GridView::\$showPrimaryKey**

*bool* = false [[line 115](#)]

**Set to display primary key in view, Default is false**

- **Access** protected

**GridView::\$sortBy**

*string* = " [[line 163](#)]

**current grid sorting field**

- **Access** protected

**GridView::\$sortOrder**

*bool* = " [[line 169](#)]

**current grid sorting order**

- **Access** protected

## GridView::\$valueField

*string* = " [[line 145](#)]

**specifies the value field**

- **Access** protected

Constructor *void* function GridView::\_\_construct(\$controlId) [[line 250](#)]

**Function Parameters:**

- *string* **\$controlId** Control Id

**sets the controlId and prepares the control attributes**

- **Access** public

*void* function GridView::addColumn(&\$column, \$column) [[line 478](#)]

**Function Parameters:**

- [GridViewColumn](#) **\$column** GridView column
- [GridViewColumn](#) **&\$column**

**add a bound column**

- **Access** public

*GridViewColumn* function *GridView::findColumn(\$dataField)* [[line 466](#)]

**Function Parameters:**

- *string* **\$dataField** data field

**returns GridViewColumn if column is found in Collection**

- **Access** public

*DomObject* function *GridView::getDomObject()* [[line 502](#)]

**returns a DomObject representing control**

- **Access** public

*DomObject* function *GridView::getPagination()* [[line 1461](#)]

**generic method for handling the table pagination**

- **Access** protected

*DomObject* function *GridView::getRowBody(\\&\$ds, \$ds)* [[line 1192](#)]

**Function Parameters:**

- [DataSet](#) **\$ds** DataSet object with current resultset
- [DataSet](#) **\\&\$ds**

**generic method for handling the row body**

- **Access** protected

*DomObject* function GridView::getRowFilter() [*line [1060](#)*]  
**generic method for handling the row header**

- **Access** protected

*DomObject* function GridView::getRowFooter() [*line [1413](#)*]  
**generic method for handling the table footer**

- **Access** protected

*DomObject* function GridView::getRowHeader() [*line [957](#)*]  
**generic method for handling the row header**

- **Access** protected

*void* function GridView::onDataBind(0) [*line [731](#)*]  
**Function Parameters:**

- *\$default* **0** value

**bind control to data**



- **Access** protected

*bool* function GridView::onLoad() [[line 754](#)]  
**called when control is loaded**

- **Access** protected

*void* function GridView::onLoadViewState(&\$viewState, \$viewState) [[line 705](#)]  
**Function Parameters:**

- *array* **\$viewState** session data
- *array* **&\$viewState**

**read view state from session**

- **Access** protected

*void* function GridView::onPost(&\$request) [[line 909](#)]  
**Function Parameters:**

- *array* **&\$request** request data

**handle post events**

- **Access** protected

*void* function GridView::onRequest(&\$request) [[line 772](#)]

**Function Parameters:**

- *array* &\$request

**process the HTTP request array**

- **Access** protected

*void* function GridView::onSaveViewState(&\$viewState, \$viewState) [[line 921](#)]

**Function Parameters:**

- *object* \$viewState session array
- *array* &\$viewState

**write view state to session**

- **Access** protected

*void* function GridView::onUpdateAjax() [[line 939](#)]

**Event called on ajax callback**

- **Access** protected

`void function GridView::setFilterValues($datafield, $values) [line 491]`

**Function Parameters:**

- *string* **\$datafield** name of datafield
- *array* **\$values** array of values

**set filter values**

- **Access** public

## Class GridViewButton

[line [20](#)]

**Represents a GridView button**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$confirmation: confirmation message
- **Author** Darnell Shinbine

### GridViewButton::\$canFilter

*bool* = false [line [26](#)]

**specifies whether column can be filtered**

- **Access** protected

### GridViewButton::\$confirmation

*string* = " [line [32](#)]

### confirmation message

- **Access** protected

Constructor *void* function GridViewButton::\_\_construct(\$dataField, [\$buttonName = "], [\$parameter = "], [\$confirmation = "], [\$headerText = "], [\$footerText = "], [\$className = "]) [line [50](#)]

#### **Function Parameters:**

- *string* **\$dataField** field name
- *string* **\$buttonName** name of button
- *string* **\$parameter** parameter
- *string* **\$headerText** header text
- *string* **\$footerText** footer text
- *string* **\$className** css class name
- **\$confirmation**

- **Access** public

*string* function GridViewButton::getItemText(\$dataField, \$parameter, \$params) [line [127](#)]

#### **Function Parameters:**

- *string* **\$dataField**

- *string* **\$parameter**
- *string* **\$params**

### get item text

- **Access** protected

*void* function GridViewButton::onPost(&\$request) [*line* [108](#)]

#### **Function Parameters:**

- *array* **&\$request** request data

### handle post events

- **Access** public

## Class GridViewCheckBox

[*line* [18](#)]

**Represents a GridView CheckBox**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

*string* function GridViewCheckBox::getItemText(\$dataField, \$parameter, \$params) [*line* [28](#)]

**Function Parameters:**

- *string* **\$dataField**
- *string* **\$parameter**
- *string* **\$params**

**get item text**

- **Access** protected

## Class GridViewColumn

[*line* [30](#)]

**Represents a GridView column**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** *string* \$dataField: datefield
- **Property** *string* \$headerText: header text
- **Property** *string* \$itemText: item text (templating allowed)
- **Property** *string* \$footerText: footer text
- **Property** *string* \$className: class name
- **Property** [EventCollection](#) \$events: event collection

- **Property** bool \$canFilter: specifies whether column can be filtered
- **Property** string \$onmouseover:
- **Property** string \$onmouseout:
- **Property** string \$onclick:
- **Property** string \$ondblclick:

#### **GridViewColumn::\$canFilter**

*bool* = true [[line 72](#)]

**specifies whether column can be filtered**

- **Access** protected

#### **GridViewColumn::\$className**

*string* = " [[line 60](#)]

**class name**

- **Access** protected

#### **GridViewColumn::\$dataField**

*string* = " [[line 36](#)]

**datefield**

- **Access** protected

### **GridViewColumn::\$events**

*EventCollection* = null [[line 66](#)]

#### **event collection**

- **Access** protected

### **GridViewColumn::\$footerText**

*string* = " [[line 54](#)]

#### **footer text**

- **Access** protected

### **GridViewColumn::\$headerText**

*string* = " [[line 42](#)]

#### **header text**

- **Access** protected

### **GridViewColumn::\$itemText**

*string* = " [[line 48](#)]

#### **item text**



- **Access** protected

#### **GridViewColumn::\$onclick**

*string* = " [[line 90](#)]

**specifies the action to take on click events**

- **Access** protected

#### **GridViewColumn::\$ondblclick**

*string* = " [[line 96](#)]

**specifies the action to take on double click events**

- **Access** protected

#### **GridViewColumn::\$onmouseout**

*string* = " [[line 84](#)]

**specifies the action to take on onmouseout events**

- **Access** protected

#### **GridViewColumn::\$onmouseover**

*string* = " [[line 78](#)]

**specifies the action to take on mouseover events**

- **Access** protected

Constructor *void* function GridViewColumn::\_\_construct(\$dataField, [\$headerText = "], [\$itemText = "], [\$footerText = "], [\$className = "]) [[line 107](#)]

**Function Parameters:**

- *string* **\$dataField** name of data field to bind column to
- *string* **\$headerText** column header text
- *string* **\$itemText** column item text (templating allowed)
- *string* **\$footerText** column footer text
- *string* **\$className** column css class name

- **Access** public

*string* function GridViewColumn::getRequestData() [[line 332](#)]

**creates string of all vars currently in the request object seperated by an '&' used to preserve application state**

- **Access** protected

*void* function GridViewColumn::onLoad() [[line 280](#)]

**handle load events**

- **Access** public

*void* function GridViewColumn::onPost(&\$request) [*line* [298](#)]

**Function Parameters:**

- *array* **&\$request** request data

## handle post events

- **Access** public

*void* function GridViewColumn::onRender() [*line* [306](#)]

## handle render events

- **Access** public

*void* function GridViewColumn::onRequest(&\$request) [*line* [289](#)]

**Function Parameters:**

- *array* **&\$request** request data

## handle request events

- **Access** public

# Class GridViewColumnAjaxPostEvent

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function GridViewColumnAjaxPostEvent::\_\_construct() [[line 26](#)]

**Constructor**

- **Access** public

# Class GridViewColumnAjaxPostEventHandler

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function GridViewColumnAjaxPostEventHandler::\_\_construct(\$callback) [[line 27](#)]

**Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class GridViewColumnCollection

[[line 19](#)]

**Represents a Collection of GridViewColumn objects**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

*bool* function GridViewColumnCollection::add(\$item) [[line 82](#)]

**Function Parameters:**

- [\*GridViewColumn\*](#) **\$item**

## add GridViewColumn to Collection

- **Access** public

*bool* function GridViewColumnCollection::addBefore(\$item, \$datafield) [*line* [53](#)]

**Function Parameters:**

- [GridViewColumn](#) \$item
- *string* \$datafield

**add GridViewColumn to Collection before**

- **Access** public

*int* function GridViewColumnCollection::findColumn(\$dataField, \$controlId) [*line* [172](#)]

**Function Parameters:**

- *string* \$controlId control id
- \$dataField

**returns index if value is found in collection**

- **Access** public

*GridViewColumn* function GridViewColumnCollection::itemAt(\$index) [*line* [102](#)]

**Function Parameters:**

- *int* \$index index of GridViewColumn

**return GridViewColumn at a specified index**

- **Access** public

*void* function GridViewColumnCollection::onLoad() [*line* [113](#)]

**handle load events**

- **Access** public

*void* function GridViewColumnCollection::onPost(&\$request) [*line* [143](#)]

**Function Parameters:**

- *array* **&\$request** request data

**handle post events**

- **Access** public

*void* function GridViewColumnCollection::onRender() [*line* [157](#)]

**handle render events**

- **Access** public

*void* function GridViewColumnCollection::onRequest(&\$request) [*line* [128](#)]

**Function Parameters:**

- `array &$request` request data

## handle request events

- **Access** public

# Class GridViewColumnPostEvent

[line [19](#)]

Provides event handling

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function GridViewColumnPostEvent::\_\_construct() [line [26](#)]

## Constructor

- **Access** public



# Class GridViewColumnPostEventHandler

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function GridViewColumnPostEventHandler::\_\_construct(\$callback) [[line 27](#)]

**Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class GridViewControlBase

[[line 21](#)]

**Represents a GridView control**

- **Package** PHPRum
- **Sub-Package** Web

- **Property** bool \$ajaxPostBack: specifies whether to perform ajax postback on change, Default is false
- **Property** bool \$escapeOutput: Specifies whether to escape the output
- **Abstract Element**

#### GridViewControlBase::\$ajaxPostBack

*string* = false [[line 39](#)]

**event request parameter**

- **Access** protected

#### GridViewControlBase::\$escapeOutput

*bool* = true [[line 45](#)]

**determines whether to escape the output**

- **Access** protected

#### GridViewControlBase::\$parameter

*string* = " [[line 27](#)]

**event request parameter**

- **Access** protected

#### GridViewControlBase::\$pkey

*string* = " [\[line 33\]](#)

## primary key

- **Access** protected

Constructor *void* function GridViewControlBase::\_\_construct(\$dataField, \$pkey, [\$parameter = "], [\$headerText = "], [\$footerText = "], [\$className = "], \$value) [\[line 76\]](#)

### **Function Parameters:**

- *string* **\$dataField** field name
- *string* **\$pkey** primary key
- *string* **\$value** value of Control
- *string* **\$parameter** parameter
- *string* **\$headerText** header text
- *string* **\$footerText** footer text
- *string* **\$className** css class name

- **Access** public

*string* function GridViewControlBase::escape(\$string) [\[line 208\]](#)

### **Function Parameters:**

- *type* **\$string** string to escape

## escape

- **Access** protected

*string* function GridViewControlBase::getItemText(\$dataField, \$parameter, \$params) [*line* [229](#)]

**Function Parameters:**

- *string* **\$dataField**
- *string* **\$parameter**
- *string* **\$params**

## get item text

- **Abstract Element**
- **Access** protected

*void* function GridViewControlBase::onPost(&\$request) [*line* [171](#)]

**Function Parameters:**

- *array* **&\$request** request data

## handle post events

- **Access** public

*void* function GridViewControlBase::onRender(&\$request) [*line* [195](#)]

**Function Parameters:**

- *array* **&\$request** request data

## handle request events

- **Access** public

*void* function GridViewControlBase::onRequest(&\$request) [*line* [154](#)]

### **Function Parameters:**

- *array* **&\$request** request data

## handle request events

- **Access** public

# Class GridViewDropDownMenu

[*line* [18](#)]

**Represents a GridView DropDownMenu**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

## GridViewDropDownMenu::\$items

*array* = [line [24](#)]

### items

- **Access** protected

Constructor *void* function GridViewDropDownMenu::\_\_construct(\$dataField, \$pkey, \$values, [\$parameter = "], [\$headerText = "], [\$footerText = "], [\$className = "], \$value) [line [38](#)]

#### **Function Parameters:**

- *string* **\$dataField** field name
- *string* **\$pkey** primary key
- *array* **\$values** list values
- *string* **\$value** value of Control
- *string* **\$parameter** parameter
- *string* **\$headerText** header text
- *string* **\$footerText** footer text
- *string* **\$className** css class name

- **Access** public

*string* function GridViewDropDownMenu::getItemText(\$dataField, \$parameter, \$params) [line [53](#)]

#### **Function Parameters:**

- *string* **\$dataField**
- *string* **\$parameter**
- *string* **\$params**

## get item text

- **Access** protected

# Class GridViewImage

[[line 18](#)]

**Represents a GridView image**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function GridViewImage::\_\_construct(\$dataField, [\$headerText = "], [\$src = "], [\$alt = "\\"], [\$footerText = "], [\$className = "]) [[line 29](#)]

### **Function Parameters:**

- *string* **\$dataField** field name
- *string* **\$headerText** header text
- *string* **\$src** src (templating allowed)
- *string* **\$alt** alt text
- *string* **\$footerText** footer text
- *string* **\$className** css class name

- **Access** public

## Class GridViewLink

[[line 18](#)]

**Represents a GridView link**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function GridViewLink::\_\_construct(\$dataField, [\$headerText = "], [\$url = "], [\$title = "], [\$target = "], [\$alt = "], [\$footerText = "], [\$className = "]) [[line 31](#)]

### **Function Parameters:**

- *string* **\$dataField** field name
- *string* **\$headerText** header text
- *string* **\$url** url (templating allowed)
- *string* **\$title** link title (templating allowed)
- *string* **\$target** link target
- *string* **\$alt** alt text
- *string* **\$footerText** footer text
- *string* **\$className** css class name



- **Access** public

## Class GridViewTextBox

[*line* [18](#)]

**Represents a GridView TextBox**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

**GridViewTextBox::\$size**

*int* = 30 [*line* [24](#)]

**size**

- **Access** protected

*string* function GridViewTextBox::getItemText(\$dataField, \$parameter, \$params) [*line* [34](#)]

**Function Parameters:**

- *string* **\$dataField**
- *string* **\$parameter**
- *string* **\$params**

**get item text**

- **Access** protected

## Class HyperLink

[[line 23](#)]

**Represents a Button Control**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$text: Specifies hyperlink text
- **Property** string \$url: Specifies hyperlink URL
- **Property** string \$target: Specifies hyperlink target
- **Property** string \$src: Specifies hyperlink image source
- **Author** Darnell Shinbine

**HyperLink::\$src**

*string* = " [[line 47](#)]

**specifies hyperlink image source**

- **Access** protected

### HyperLink::\$target

*string* = *'\_self'* [*line 41*]

**specifies hyperlink target**

- **Access** protected

### HyperLink::\$text

*string* = *''* [*line 29*]

**specifies hyperlink text**

- **Access** protected

### HyperLink::\$url

*string* = *''* [*line 35*]

**specifies hyperlink URL**

- **Access** protected

Constructor *void* function HyperLink::\_\_construct(*\$*controlId, [*\$*text = *''*], [*\$*url = *''*], [*\$*target = *'\_self'*]) [*line 57*]

#### **Function Parameters:**

- *string* ***\$*controlId** Control Id
- *string* ***\$*text** Button text
- ***\$*url**

- **\$target**

## Constructor

- **Access** public

*DomObject* function `HyperLink::getDomObject()` [*line* [149](#)]

### **getDomObject**

returns a DomObject representing control

- **Access** public

## Class InputAjaxChangeEvent

[*line* [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shimbine

Constructor *void* function `InputAjaxChangeEvent::__construct()` [*line* [26](#)]

### **Constructor**

- **Access** public

## Class InputAjaxChangeEventHandler

[line [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function InputAjaxChangeEventHandler::\_\_construct(\$callback) [line [27](#)]

**Function Parameters:**

- *string* **\$callback** call back

### Constructor

- **Access** public

# Class InputAjaxPostEvent

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function InputAjaxPostEvent::\_\_construct() [[line 26](#)]

**Constructor**

- **Access** public

# Class InputAjaxPostEventHandler

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function InputAjaxPostEventHandler::\_\_construct(\$callback) [[line 27](#)]

**Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class InputBase

[line [33](#)]

**Provides base functionality for Input Controls**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$defaultHTMLControlId: Specifies the id of the default html control
- **Property** string \$dataField: Name of the data field in the datasource
- **Property** bool \$autoPostBack: Specifies whether form will perform postback on change, Default is false
- **Property** bool \$ajaxPostBack: specifies whether to perform ajax postback on change, Default is false
- **Property** bool \$ajaxValidation: specifies whether to perform ajax validation, Default is false
- **Property** bool \$readonly: Specifies whether control is readonly
- **Property** bool \$disabled: Specifies whether the control is disabled
- **Property** string \$label: Specifies control label
- **Property** string \$tooltip: Specifies control tooltip
- **Property** int \$tabIndex: Specifies the tab order if the control

- **Property** string \$value: Gets or sets value of control
- **Property** bool \$submitted: Specifies whether the data has been submitted
- **Property** bool \$changed: Specifies whether the data has been changed
- **Property** array \$validators: Array of validators
- **Author** Darnell Shinbine
- **Abstract Element**

#### **InputBase::\$ajaxPostBack**

*bool* = false [[line 63](#)]

**Specifies whether form will submit ajax postback on change, Default is false**

- **Access** protected

#### **InputBase::\$ajaxValidation**

*bool* = false [[line 69](#)]

**Specifies whether form will submit ajax validation, Default is false**

- **Access** protected

#### **InputBase::\$autoPostBack**

*bool* = false [[line 57](#)]

**Specifies whether form will submit postback on change, Default is false**

- **Access** protected



#### **InputBase::\$changed**

*bool* = false [[line 117](#)]

**specifies whether the control has changed**

- **Access** protected

#### **InputBase::\$dataField**

*string* = "" [[line 51](#)]

**Name of the data field in the datasource**

- **Access** protected

#### **InputBase::\$defaultHTMLControlId**

*string* = "" [[line 45](#)]

**specifies the id of the default html control**

- **Access** protected

#### **InputBase::\$disabled**

*bool* = false [[line 81](#)]

**Specifies whether the control is disabled, Default is false**

- **Access** protected

#### **InputBase::\$enableViewState**

*bool* = false [[line 39](#)]

**specifies whether the server control persists its view state, Default is true**

- **Access** protected

#### **InputBase::\$label**

*string* = " [[line 87](#)]

**Specifies control label**

- **Access** protected

#### **InputBase::\$readonly**

*bool* = false [[line 75](#)]

**Specifies whether the control is readonly, Default is false**

- **Access** protected

#### **InputBase::\$submitted**

*bool* = false [[line 111](#)]

**Specifies whether the data has been submitted**

- **Access** protected

**InputBase::\$tabIndex**

*int* = 0 [*line* [99](#)]

**Specifies the tab order if the control**

- **Access** protected

**InputBase::\$tooltip**

*string* = " [*line* [93](#)]

**specifies control tool tip**

- **Access** protected

**InputBase::\$validators**

*ValidatorCollection* = null [*line* [123](#)]

**contains a collection of validators**

- **Access** protected

**InputBase::\$value**

*string* = null [[line 105](#)]

## **Gets or sets value of control**

- **Access** protected

Constructor *void* function InputBase::\_\_construct(\$controlId, [\$default = null]) [[line 142](#)]

### **Function Parameters:**

- *string* **\$controlId** Control Id
- *string* **\$default** Default value

## **Constructor**

The constructor sets attributes based on session data, triggering events, and is responsible for formatting the proper request value and garbage handling

- **Access** public

*void* function InputBase::addValidator(\\\$validator) [[line 315](#)]

### **Function Parameters:**

- [ValidatorBase](#) \\\$validator

## **adds a validator to the control**

- **Access** public

*void function* InputBase::error([\$args = array()]) [*line* [356](#)]

***Function Parameters:***

- *array* **\$args** parameters

**renders error message if control does not validate**

- **Access** public

*void function* InputBase::fetchError([\$args = array()]) [*line* [368](#)]

***Function Parameters:***

- *array* **\$args** parameters

**returns the error message element**

- **Access** public

*void function* InputBase::fillDataSource(\&\$ds, \$ds) [*line* [397](#)]

***Function Parameters:***

- *ArrayAccess* **\$ds** data source to fill
- *ArrayAccess* **\&\$ds**

**update data source with Control value**

- **Access** public

*bool* function InputBase::focus() [*line* [303](#)]  
**sets focus to the control**

- **Access** public

*DomObject* function InputBase::getDomObject() [*line* [385](#)]  
**returns an input DomObject representing control**

- **Access** public

*DomObject* function InputBase::getInputDomObject() [*line* [429](#)]  
**returns an input DomObject representing control**

- **Access** protected

*bool* function InputBase::onDataBind() [*line* [502](#)]  
**bind control to datasource gets record from dataobject and sets the control value to datafield value**

- **Access** protected

*void* function InputBase::onInit() [[line 516](#)]  
**called when control is initiated**

- **Access** protected

*void* function InputBase::onLoad() [[line 527](#)]  
**called when control is loaded**

- **Access** protected

*void* function InputBase::onLoadViewState(&\$viewState) [[line 484](#)]  
**Function Parameters:**

- *array* **&\$viewState** session array

**read view state from session**

- **Access** protected

*void* function InputBase::onPost(&\$request) [[line 578](#)]  
**Function Parameters:**

- *array* **&\$request** request data

**handle post events**

- **Access** protected

*void function* InputBase::onRequest(&\$request) [*line* [539](#)]

***Function Parameters:***

- *array* **&\$request** request data

**process the HTTP request array**

- **Access** protected

*void function* InputBase::onSaveViewState(&\$viewState) [*line* [608](#)]

***Function Parameters:***

- *array* **&\$viewState** session array

**write view state to session**

- **Access** protected

*void function* InputBase::onUpdateAjax() [*line* [622](#)]

**Event called on ajax callback**

- **Access** protected



*void* function InputBase::readDataSource(\&\$ds, \$ds) [[line 412](#)]

**Function Parameters:**

- *ArrayAccess* **\$ds** data source to read
- *ArrayAccess* **\&\$ds**

**update Control value with data from the data source**

- **Access** public

*bool* function InputBase::validate([\&\$errMsg = "", [\&\$controlToFocus = null], \$errMsg) [[line 327](#)]

**Function Parameters:**

- *string* **\$errMsg** error message
- **&\$errMsg**
- [InputBase](#) **&\$controlToFocus**

**validates control data, returns true on success**

- **Access** public

## Class InputChangeEvent

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function InputChangeEvent::\_\_construct() [[line 26](#)]

### **Constructor**

- **Access** public

## Class InputChangeEventHandler

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function InputChangeEventHandler::\_\_construct(\$callback) [[line 27](#)]

### **Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

## Class InputPostEvent

[*line* [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function InputPostEvent::\_\_construct() [*line* [26](#)]

## Constructor

- **Access** public

## Class InputPostEventHandler

[*line* [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function InputPostEventHandler::\_\_construct(\$callback) [[line 27](#)]

**Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class JSONServiceBase

[[line 23](#)]

**This class handles all remote procedure calls for a web service**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$encoding: specifies the character set for the soap message, default is ISO-8859-1
- **Property** string \$cache: specifies whether to cache the WSDL, default is WSDL\_CACHE\_NONE

- **Property** string \$version: specifies the SOAP version, default is SOAP\_1\_2
- **Property** string \$namespace: specifies the namespace, default is controller id
- **Author** Darnell Shinbine
- **Abstract Element**

## JSONServiceBase::\$options

*string* = null [*line 29*]

**specifies encoding options**

- **Access** protected

*void* function JSONServiceBase::configure() [*line 85*]

**configure the web service**

- **Access** protected

*void* function JSONServiceBase::handle(\\&\$request, &\$request) [*line 97*]

**Function Parameters:**

- [\*HttpRequest\*](#) &\$request *HttpRequest* object
- [\*HttpRequest\*](#) \\&\$request

**this method will handle the web service request**

- **Access** public

# Class Label

[*line* [20](#)]

## Represents a Button Control

- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$text: Specifies button text
- **Author** Darnell Shinbine

### Label::\$text

*string* = " [*line* [26](#)]

### specifies button text

- **Access** protected

Constructor *void* function Label::\_\_construct(\$controlId, [\$text = "]) [*line* [36](#)]

#### **Function Parameters:**

- *string* **\$controlId** Control Id
- *string* **\$text** Button text

### Constructor

- **Access** public

*DomObject* function `Label::getDomObject()` [[line 108](#)]

### **getDomObject**

returns a DomObject representing control

- **Access** public

## Class `ListBox` [[line 20](#)]

**Represents a ListBox Control**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** `int $listSize`: Size of listbox
- **Author** Darnell Shinbine

### **`ListBox::$listSize`**

*int* = 6 [[line 26](#)]

**Size of listbox, default is 6**

- **Access** protected

*DomObject* function `ListBox::getDomObject()` [[line 75](#)]  
**returns a DomObject representing control**

- **Access** public

*void* function `ListBox::onUpdateAjax()` [[line 162](#)]  
**Event called on ajax callback**

- **Access** protected

## Class `ListControlBase` [[line 23](#)]

**Provides base functionality for List Controls**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** [ListItemCollection](#) \$items: Collection of list items
- **Property** bool \$multiple: Specifies whether multiple selections are allowed
- **Property** string \$textField: Specifies name of value text in datasource
- **Property** string \$valueField: Specifies name of value field in datasource



- **Author** Darnell Shinbine
- **Abstract Element**

#### **ListControlBase::\$items**

*ListItemCollection* = [line [29](#)]

**collection of items**

- **Access** protected

#### **ListControlBase::\$multiple**

*bool* = false [line [35](#)]

**Specifies whether multiple selections are allowed**

- **Access** protected

#### **ListControlBase::\$textField**

*string* = " [line [41](#)]

**Specifies name of text field in datasource**

- **Access** protected

#### **ListControlBase::\$valueField**

*string* = " [line [47](#)]

**Specifies name of value field in datasource**

- **Access** protected

Constructor *void* function ListControlBase::\_\_construct(\$controlId, [\$default = null]) [[line 57](#)]

**Function Parameters:**

- *string* **\$controlId** Control Id
- *string* **\$default** Default value

## Constructor

- **Access** public

*void* function ListControlBase::onDataBind(0) [[line 182](#)]

**Function Parameters:**

- *\$default* **0** value

## bind control to data

- **Access** protected

*bool* function ListControlBase::onLoad() [[line 213](#)]

**called when control is loaded**

- **Access** protected

*void* function ListControlBase::onLoadViewState(&\$viewState, \$viewState) [*line* [165](#)]

**Function Parameters:**

- *object* **\$viewState** session array
- *array* **&\$viewState**

**read view state from session**

- **Access** protected

*void* function ListControlBase::onRequest(&\$request) [*line* [226](#)]

**Function Parameters:**

- *array* **&\$request**

**process the HTTP request array**

- **Access** protected

*bool* function ListControlBase::validate([&\$errMsg = "], [&\$controlToFocus = null], \$errMsg) [*line* [139](#)]

**Function Parameters:**

- *string* **\$errMsg** error message
- **&\$errMsg**
- [InputBase](#) **&\$controlToFocus**

validates control data, returns true on success

- **Access** public

## Class ListItemCollection

[[line 19](#)]

Represents a Collection of list items

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

void function ListItemCollection::add(\$key, \$value) [[line 28](#)]

**Function Parameters:**

- *mixed* **\$key** key
- *mixed* **\$value** value

Adds a new key/item pair to a Dictionary object if key does not already exist

- **Access** public

*bool* function ListItemCollection::containsItem(\$value) [*line* [40](#)]

**Function Parameters:**

- *mixed* **\$value** value

**Returns a Boolean value that indicates whether a specified value exists in the Dictionary object**

- **Access** public

*int* function ListItemCollection::indexOf(\$key) [*line* [59](#)]

**Function Parameters:**

- *mixed* **\$key** key

**returns index of key found in Dictionary**

- **Access** public

*int* function ListItemCollection::indexOfItem(\$value, \$key) [*line* [79](#)]

**Function Parameters:**

- *mixed* **\$key** key
- **\$value**

**returns index of item found in Dictionary**

- **Access** public

## Class LoginForm

[line [32](#)]

**Represents a LoginForm**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$invalidCredentialsMsg: Specifies the message when login failed
- **Property** string \$disabledMsg: Specifies the message when account is disabled
- **Property** string \$lockoutMsg: Specifies the message when account is locked out
- **Property** string \$emailNotFoundMsg: Specifies the message when email address is not found
- **Property** string \$passwordResetSentMsg: Specifies the message when password reset has been sent
- **Property** string \$passwordResetMsg: Specifies the message when password has been reset
- **Property** string \$successMsg: Specifies the success message
- **Property** string \$emailFromAddress: Specifies the from address of the email message
- **Property** string \$emailSubject: Specifies the subject line of the email message
- **Property** [IMailClient](#) \$mailClient: Specifies the mail client to send the message with
- **Property** [TextBox](#) \$username: username control
- **Property** [TextBox](#) \$password: password control
- **Property** [TextBox](#) \$new\_password: new password control when resetting password
- **Author** Darnell Shinbine

#### **LoginForm::\$disabledMsg**

*string* = 'Account is suspended, please contact the system administrator' [[line 44](#)]

**specifies the message when account is disabled**

- **Access** protected

#### **LoginForm::\$emailFromAddress**

*string* = 'no-reply@localhost.local' [[line 80](#)]

**Specifies the from address when email is sent**

- **Access** protected

#### **LoginForm::\$emailNotFoundMsg**

*string* = 'No account with that email address was found' [[line 56](#)]

**specifies the message when email address does not exist**

- **Access** protected

#### **LoginForm::\$emailSubject**

*string* = 'Password reset confirmation' [[line 86](#)]

**Specifies the subject line when email is sent**

- **Access** protected

#### **LoginForm::\$invalidCredentialsMsg**

*string* = 'Invalid credentials provided' [[line 38](#)]

**specifies the message when login failed**

- **Access** protected

#### **LoginForm::\$lockoutMsg**

*string* = 'Account is locked out due to too many attempts, please try again later' [[line 50](#)]

**specifies the message when account is locked out**

- **Access** protected

#### **LoginForm::\$mailClient**

*IMailClient* = [[line 92](#)]

**Specifies the mail client for sending**

- **Access** protected

#### **LoginForm::\$passwordResetMsg**

*string* = 'Your account password has been reset' [[line 68](#)]

**specifies the message when email address does not exist**



- **Access** protected

#### **LoginForm::\$passwordResetSentMsg**

*string* = 'Your account password has been sent to the email address specified' [[line 62](#)]

**specifies the message when email address does not exist**

- **Access** protected

#### **LoginForm::\$successMsg**

*string* = 'You have successfully logged in' [[line 74](#)]

**specifies the message on successful login**

- **Access** protected

Constructor *void* function LoginForm::\_\_construct(\$controlId) [[line 102](#)]

#### **Function Parameters:**

- *string* **\$controlId** Control Id

#### **Constructor**

- **Access** public

*DomObject* function LoginForm::getDomObject() [[line 502](#)]  
**returns a DomObject representing control**

- **Access** public

*void* function LoginForm::onInit() [[line 266](#)]  
**called when control is initiated**

- **Access** protected

*void* function LoginForm::onRequest(&\$request) [[line 318](#)]  
**Function Parameters:**

- *array* **&\$request** request array

**process the HTTP request array**

- **Access** protected

## Class LoginFormSubmitEvent

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function LoginFormSubmitEvent::\_\_construct() [[line 26](#)]

### **Constructor**

- **Access** public

## Class LoginFormSubmitEventHandler

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function LoginFormSubmitEventHandler::\_\_construct(\$callback) [[line 27](#)]

### **Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class MasterView

[line [18](#)]

**Represents a MasterView Control**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shimbine

Constructor *void* function MasterView::\_\_construct(\$controlId) [line [35](#)]

**Function Parameters:**

- *string* **\$controlId** Control Id

## Constructor

sets the controlId and prepares the control attributes

- **Access** public

*void* function MasterView::content() [line [65](#)]

## render content

- **Access** public

*string* function MasterView::getContentAreaIdString() [*line* [76](#)]

### **get content area id string**

- **Access** public

*void* function MasterView::handleMasterViewPostEvents(&\$request) [*line* [161](#)]

#### **Function Parameters:**

- *array* **&\$request** request data

## handle post events

- **Access** public

*void* function MasterView::initMasterView() [*line* [87](#)]

### **called when control is initiated**

- **Access** public

*void* function MasterView::loadMasterView() [*line* [125](#)]

### **called when all controls are loaded**

- **Access** public

*void* function MasterView::loadMasterViewViewState(&\$viewState) [*line* [106](#)]

***Function Parameters:***

- *array* **&\$viewState** session data

**read view state from session**

- **Access** public

*void* function MasterView::masterViewRequestProcessor(&\$request) [*line* [144](#)]

***Function Parameters:***

- *array* **&\$request** request data

**process the HTTP request array**

- **Access** public

*void* function MasterView::saveMasterViewViewState(&\$viewState) [*line* [178](#)]

***Function Parameters:***

- *array* **&\$viewState** session data

**write view state to session**

- **Access** public

## Class MasterViewInitEvent

[*line* [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shimbine

Constructor *void* function MasterViewInitEvent::\_\_construct() [*line* [26](#)]

**Constructor**

- **Access** public

## Class MasterViewInitEventHandler

[*line* [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function MasterViewInitEventHandler::\_\_construct(\$callback) [*line* [27](#)]

**Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class MasterViewLoadEvent

[*line* [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine



Constructor *void* function MasterViewLoadEvent::\_\_construct() [[line 26](#)]

## Constructor

- **Access** public

# Class MasterViewLoadEventHandler

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shimbine

Constructor *void* function MasterViewLoadEventHandler::\_\_construct(\$callback) [[line 27](#)]

## **Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class Page

[line [20](#)]

## Represents an Page Control

- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$onload: Specifies jscript events for page load
- **Author** Darnell Shinbine

### Page::\$contentType

*string* = 'text/html' [line [26](#)]

#### specifies content-Type

- **Access** protected

### Page::\$linkElements

*array* = array() [line [50](#)]

#### array of link elements

- **Access** protected

### Page::\$metaElements

*array* = array() [line [44](#)]

## array of meta elements

- **Access** protected

### Page::\$ObjectElements

*array* = array() [*line* [56](#)]

## array of object elements

- **Access** protected

### Page::\$onload

*string* = " [*line* [32](#)]

## specify javascript to execute on page load

- **Access** protected

### Page::\$scriptElements

*array* = array() [*line* [38](#)]

## array of script elements

- **Access** protected

Constructor *void* function Page::\_\_construct(\$controlId) [[line 77](#)]

**Function Parameters:**

- *string* **\$controlId** Control Id

**sets the controlId and prepares the control attributes**

- **Access** public

*int* function Page::addLink(\$href, [\$rel = 'stylesheet'], [\$type = 'text/css'], [\$media = 'all'], [\$charset = 'iso-8859-1']) [[line 201](#)]

**Function Parameters:**

- *string* **\$href** source
- *string* **\$rel** relationship, default is stylesheet
- *string* **\$type** type, default is text/css
- *string* **\$media** media, default is 'all'
- *string* **\$charset** character set, default is iso-8859-1

**add link element**

- **Access** public

*int* function Page::addMetaTag(\$name, \$content) [[line 140](#)]

**Function Parameters:**

- *string* **\$name** the meta tag name to add
- *string* **\$content** the meta tag content value to add

## add metatag

- **Access** public

*int* function Page::addScript(\$src, [\$type = 'text/javascript'], [\$language = 'Javascript'], [\$charset = 'utf-8']) [*line* [169](#)]

### **Function Parameters:**

- *string* **\$src** source
- *string* **\$type** type, default is text/javascript
- *string* **\$language** script language, default is 'Javascript'
- *string* **\$charset** character set, default is iso-8859-1

## add script element

- **Access** public

*void* function Page::clearAjaxBuffer() [*line* [266](#)]

### **clear ajax output buffer**

- **Access** public

*string* function Page::fetch([\$args = array()]) [*line* [289](#)]

### **Function Parameters:**

- *array* **\$args** widget parameters

**returns an html string**

- **Access** public

*string* function Page::fetchPageHeader() [[line 351](#)]

**return page header**

- **Access** protected

*string* function Page::getAjaxBuffer() [[line 253](#)]

**return ajax output buffer as a string**

- **Access** public

*bool* function Page::isAjaxBufferReady() [[line 277](#)]

**specifies whether the buffer contains output**

- **Access** public

*void* function Page::loadAjaxBuffer(\$output) [[line 230](#)]

**Function Parameters:**

- *string* **\$output** output content

## load content into ajax output buffer

- **Access** protected

*void* function Page::loadAjaxJScriptBuffer(\$javascript) [*line* [242](#)]

### **Function Parameters:**

- *string* **\$javascript** output javascript content

## load content into ajax jscript output buffer

- **Access** public

*void* function Page::onInit() [*line* [311](#)]

### **Event called when control is initiated**

- **Access** protected

*void* function Page::onPost(&\$request) [*line* [339](#)]

### **Function Parameters:**

- *array* **&\$request** request data

## handle post events

- **Access** protected

*void* function Page::onRequest(&\$request) [*line* [326](#)]

**Function Parameters:**

- *array* **&\$request** request data

**Event called when request is processed**

- **Access** protected

## Class PageControllerBase

[*line* [27](#)]

**This class handles all requests for a specific page. In addition provides access to a Page component to manage any WebControl components.**

The PageControllerBase exposes 4 protected properties

- **Package** PHPRum
- **Sub-Package** Web
- **Property** [Page](#) \$page: Contains an instance of the Page component
- **Property** string \$theme: Specifies the theme for this page

2 Additional public readonly properties are exposed



- **Property** bool \$isPostBack: Specifies whether the request is a postback
- **Property** bool \$isAjaxPostBack: Specifies whether the request is an ajax postback
- **Author** Darnell Shinbine
- **Abstract Element**

**PageControllerBase::\$page**

*Page = null* [*line 33*]

**Contains an instance of the Page component**

- **Access** protected

**PageControllerBase::\$theme**

*string = "* [*line 39*]

**Specifies the theme for this page**

- **Access** protected

View function PageControllerBase::getView(\\&\$request, &\$request) [*line 140*]

**Function Parameters:**

- [\*HttpRequest\*](#) &\$request HttpRequest object
- [\*HttpRequest\*](#) \\&\$request

**return view component for rendering**

- **Access** public

*void* function PageControllerBase::onLoad() [[line 127](#)]

**Event called when object created**

- **Access** protected

## Class PagePostEvent

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function PagePostEvent::\_\_construct() [[line 26](#)]

**Constructor**

- **Access** public

# Class PagePostEventHandler

[line [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function PagePostEventHandler::\_\_construct(\$callback) [line [27](#)]

**Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class PageRequestEvent

[line [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web

- **Author** Darnell Shinbine

Constructor *void* function PageRequestEvent::\_\_construct() [[line 26](#)]

## **Constructor**

- **Access** public

# Class PageRequestEventHandler

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function PageRequestEventHandler::\_\_construct(\$callback) [[line 27](#)]

## **Function Parameters:**

- *string* **\$callback** call back

## **Constructor**

- **Access** public

# Class RadioButton

[[line 20](#)]

**Represents a RadioButton Control**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** bool \$checked: Specifies whether button is checked
- **Author** Darnell Shinbine

**RadioButton::\$checked**

*bool* = false [[line 26](#)]

**Specifies checked status**

- **Access** protected

Constructor *void* function RadioButton::\_\_construct(\$controlId, [\$default = false]) [[line 39](#)]

**Function Parameters:**

- *string* **\$controlId** Control Id
- *string* **\$default** Default value

## Constructor

The constructor sets attributes based on session data, triggering events, and is

responsible for formatting the proper request value and garbage handling

- **Access** public

*DomObject* function `RadioButton::getDomObject()` [[line 93](#)]  
**returns a DomObject representing control**

- **Access** public

*bool* function `RadioButton::onLoad()` [[line 119](#)]  
**called when control is loaded**

- **Access** protected

*void* function `RadioButton::onRequest(&$request)` [[line 132](#)]  
**Function Parameters:**

- *array* **&\$request**

**process the HTTP request array**

- **Access** protected

# Class RadioButtonList

[[line 21](#)]

**Represents a RadioButtonList Control**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$onclick: Specifies the action to take on click events
- **Property** string \$ondblclick: Specifies the action to take on double click events
- **Author** Darnell Shinbine

**RadioButtonList::\$multiple**

*mixed* = false [[line 36](#)]

**Size of listbox**

- **Access** protected

**RadioButtonList::\$onclick**

*mixed* = " [[line 26](#)]

**Specifies the action to take on click events**

- **Access** protected

## RadioButtonList::\$ondblclick

*mixed* = " [\[line 31\]](#)

**Specifies the action to take on double click events**

- **Access** protected

*DomObject* function RadioButtonList::getDomObject() [\[line 101\]](#)

**returns a DomObject representing control**

- **Access** public

*bool* function RadioButtonList::onLoad() [\[line 184\]](#)

**called when control is loaded**

- **Access** protected

## Class RadioGroup

[\[line 18\]](#)

**Represents a RadioGroup Control**

- **Package** PHPRum
- **Sub-Package** Web



- **Author** Darnell Shinbine

Constructor *void* function `RadioGroup::__construct($controlId, [$default = null])` [*line 30*]

**Function Parameters:**

- *string* **\$controlId** Control Id
- *string* **\$default** Default value

## Constructor

The constructor sets attributes based on session data, triggering events, and is responsible for formatting the proper request value and garbage handling

- **Access** public

*void* function `RadioGroup::add($control, &$control)` [*line 42*]

**Function Parameters:**

- [\*RadioButton\*](#) **&\$control** instance of a `RadioButton`
- [\*RadioButton\*](#) **\$control**

## adds child control to collection

- **Access** public

*DomObject* function `RadioGroup::getDomObject()` [*line 53*]

**returns a DomObject representing control**

- **Access** public

*bool* function RadioGroup::onLoad() [*line* [81](#)]  
**called when control is loaded**

- **Access** protected

*void* function RadioGroup::onRequest(&\$request) [*line* [118](#)]  
**Function Parameters:**

- *array* **&\$request** request data

**process the HTTP request array**

- **Access** protected

## Class ReportDetailsEvent

[*line* [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web

- **Author** Darnell Shinbine

Constructor *void* function ReportDetailsEvent::\_\_construct() [[line 26](#)]

## Constructor

- **Access** public

# Class ReportDetailsEventHandler

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function ReportDetailsEventHandler::\_\_construct(\$callback) [[line 27](#)]

## **Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class ReportFooterEvent

[line [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function ReportFooterEvent::\_\_construct() [line [26](#)]

**Constructor**

- **Access** public

# Class ReportFooterEventHandler

[line [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web

- **Author** Darnell Shinbine

Constructor *void* function ReportFooterEventHandler::\_\_construct(\$callback) [[line 27](#)]

**Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class ReportGroup

[[line 22](#)]

## Represents a ReportGroup

- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$groupName: Contains the group name
- **Property** [DataSet](#) \$dataSource: Contains the DataSet for the current group
- **Property** [ReportView](#) \$report: Contains the ReportView object
- **Author** Darnell Shinbine

Constructor *void* function ReportGroup::\_\_construct(&\$report, \$groupName, \\&\$ds, \$controlId) [[line 50](#)]

**Function Parameters:**

- *string* **\$controlId** Control Id
- [ReportView](#) &\$report
- **\$groupName**
- [DataSet](#) \\&\$ds

## Constructor

- **Access** public

# Class ReportGroupFooterEvent

[*line* [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shimbine

Constructor *void* function ReportGroupFooterEvent::\_\_construct([**\$groupName** = 'Group']) [*line* [27](#)]

### **Function Parameters:**

- *string* **\$groupName** group name

## Constructor

- **Access** public

## Class ReportGroupFooterEventHandler

[line [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function ReportGroupFooterEventHandler::\_\_construct(\$callback, [\$groupName = 'Group']) [line [28](#)]

### **Function Parameters:**

- *string* **\$callback** call back
- *string* **\$groupName** group name

### **Constructor**

- **Access** public

# Class ReportGroupHeaderEvent

[line [19](#)]

Provides event handling

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function ReportGroupHeaderEvent::\_\_construct([\$groupName = 'Group']) [line [27](#)]

**Function Parameters:**

- *string* **\$groupName** group name

## Constructor

- **Access** public

# Class ReportGroupHeaderEventHandler

[line [19](#)]

Provides event handling



- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function ReportGroupHeaderEventHandler::\_\_construct(\$callback, [\$groupName = 'Group']) [[line 28](#)]

**Function Parameters:**

- *string* **\$callback** call back
- *string* **\$groupName** group name

## Constructor

- **Access** public

# Class ReportHeaderEvent

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function ReportHeaderEvent::\_\_construct() [[line 26](#)]

## Constructor

- **Access** public

## Class ReportHeaderEventHandler

[*line* [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function ReportHeaderEventHandler::\_\_construct(\$callback) [*line* [27](#)]

### ***Function Parameters:***

- *string* **\$callback** call back

### **Constructor**

- **Access** public

# Class ReportView

[[line 21](#)]

**Represents a ReportView Control**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$title: Report title
- **Property** [DataSet](#) \$groupDataSource: Contains the DataSet for the current group
- **Author** Darnell Shinbine

Constructor *void* function ReportView::\_\_construct(\$controlId) [[line 62](#)]

**Function Parameters:**

- *string* **\$controlId** Control Id

## Constructor

- **Access** public

*void* function ReportView::addFilter(\$field, \$operator, \$value) [[line 190](#)]

**Function Parameters:**

- **\$field**
- **\$operator**
- **\$value**

**adds a filter to the report**

- **Access** public

*void* function ReportView::addGrouping(\$field, [\$ascending = true]) [*line* [153](#)]

***Function Parameters:***

- **\$field**
- **\$ascending**

**adds a grouping field to the report**

- **Access** public

*void* function ReportView::addSorting(\$field, [\$ascending = true]) [*line* [179](#)]

***Function Parameters:***

- **\$field**
- **\$ascending**

**adds a sorting field to the report**

- **Access** public

*string* function ReportView::fetch([\$array = array()], \$args) [*line* [206](#)]

***Function Parameters:***

- *array* **\$args** widget parameters

- *array* **\$array**

**returns an html string**

- **Access** public

*DomObject* function ReportView::getDomObject() [*line* [243](#)]  
**returns a DomObject for rendering**

- **Access** public

*void* function ReportView::getGrouping() [*line* [273](#)]  
**return grouping array**

- **Access** protected

*void* function ReportView::onDataBind(0) [*line* [255](#)]  
**Function Parameters:**

- *\$default* **0** value

**bind control to data**

- **Access** protected

# Class RESTWebServiceBase

*[line [18](#)]*

**This class handles all remote procedure calls for a REST web service**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine
- **Abstract Element**

## RESTWebServiceBase::\$options

*string = null* *[line [24](#)]*

**specifies encoding options**

- **Access** protected

*void function* RESTWebServiceBase::configure() *[line [58](#)]*

**configure the server**

- **Access** protected

`void function RESTWebServiceBase::delete($args) [line 52]`

***Function Parameters:***

- `array $args`

## handle delete requests

- **Access** public

`void function RESTWebServiceBase::get($args) [line 31]`

***Function Parameters:***

- `array $args`

## handle get requests

- **Access** public

`void function RESTWebServiceBase::handle(\\&$request, &$request) [line 84]`

***Function Parameters:***

- [\*HttpRequest\*](#) `&$request` `HttpRequest` object
- [\*HttpRequest\*](#) `\\&$request`

## this method will handle the web service request

- **Access** public

*void function RESTWebServiceBase::post(\$args) [line [38](#)]*

***Function Parameters:***

- *array \$args*

## handle post requests

- **Access public**

*void function RESTWebServiceBase::put(\$args) [line [45](#)]*

***Function Parameters:***

- *array \$args*

## handle put requests

- **Access public**

# Class SOAPWebServiceBase

*[line [23](#)]*

**This class handles all remote procedure calls for a SOAP web service**



- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$encoding: specifies the character set for the soap message, default is ISO-8859-1
- **Property** string \$cache: specifies whether to cache the WSDL, default is WSDL\_CACHE\_NONE
- **Property** string \$version: specifies the SOAP version, default is SOAP\_1\_2
- **Property** string \$namespace: specifies the namespace, default is controller id
- **Author** Darnell Shinbine
- **Abstract Element**

#### SOAPWebServiceBase::\$cache

*string* = WSDL\_CACHE\_NONE [[line 35](#)]

**specifies whether to cache the WSDL**

- **Access** protected

#### SOAPWebServiceBase::\$encoding

*string* = 'ISO-8859-1' [[line 29](#)]

**specifies the character set for the soap message**

- **Access** protected

#### SOAPWebServiceBase::\$namespace

*string* = [[line 47](#)]

**specifies the namespace**

- **Access** protected

#### **SOAPWebServiceBase::\$version**

*string* = SOAP\_1\_2 [[line 41](#)]

**specifies the SOAP version**

- **Access** protected

#### **SOAPWebServiceBase::\$xreserved\_methods**

```
array = array(  
    '__construct',  
    '__set',  
    '__get',  
    'getView',  
    'getRoles',  
    'getCached',  
    'login',  
    'configure',  
    'handle',  
    'getWSDL',  
    'getWSDLURL') [line 53]
```

**array of reserved methods**

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

*void* function SOAPWebServiceBase::configure() [[line 136](#)]

**configure the web service**

- **Access** protected

*void function SOAPWebServiceBase::handle(\\&\$request, &\$request) [line [151](#)]*

**Function Parameters:**

- [HTTPRequest](#) &\$request HTTPRequest object
- [HTTPRequest](#) \\&\$request

**this method will handle the web service request**

- **Access** public

## Class TextArea

*[line [18](#)]*

**Represents a TextBox Control**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

**TextArea::\$multiline**

*bool = true [line [24](#)]*

**Specifies whether textbox will accept multiple lines of text, default is true**

- **Access** protected

## Class TextBox

[line [28](#)]

**Represents a TextBox Control**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** bool \$multiline: Specifies whether textbox will accept multiple lines of text
- **Property** bool \$mask: Specifies whether to characters will be masked
- **Property** int \$maxLength: Specifies Max Length of value when defined
- **Property** bool \$disableAutoComplete: Specifies whether to disable the browsers auto complete feature
- **Property** bool \$disableEnterKey: Specifies whether to disable the enter key
- **Property** int \$size: Specifies the size of a textbox
- **Property** int \$rows: Specifies that number of rows in multiline textbox
- **Property** int \$cols: Specifies that number of columns in multiline textbox
- **Property** string \$watermark: Specifies an optional watermark that is displayed when the control is empty
- **Author** Darnell Shinbine

**TextBox::\$cols**

*int* = 60 [line [76](#)]

**Specifies that number of columns in multiline textbox, default is 60**

- **Access** protected

**TextBox::\$disableAutoComplete**

*bool* = false [[line 52](#)]

**Specifies whether to disable the browsers auto complete feature, default is false**

- **Access** protected

**TextBox::\$disableEnterKey**

*bool* = false [[line 58](#)]

**Specifies whether to disable the enter key, default is false**

- **Access** protected

**TextBox::\$mask**

*bool* = false [[line 40](#)]

**Specifies whether to characters will be masked, default is false**

- **Access** protected

### **TextBox::\$maxLength**

*int* = 0 [[line 46](#)]

**Max Length of value when set to non zero, default is 0**

- **Access** protected

### **TextBox::\$multiline**

*bool* = false [[line 34](#)]

**Specifies whether textbox will accept multiple lines of text, default is false**

- **Access** protected

### **TextBox::\$rows**

*int* = 5 [[line 70](#)]

**Specifies that number of rows in multiline textbox, default is 5**

- **Access** protected

### **TextBox::\$size**

*int* = 30 [[line 64](#)]

**Specifies the size of a textbox, default is 30**

- **Access** protected

**TextBox::\$watermark**

*string* = " [[line 82](#)]

**Specifies an optional watermark that is displayed when the control is empty**

- **Access** protected

*void* function TextBox::fillDataSet(\\&\$ds, \$ds) [[line 174](#)]

**Function Parameters:**

- [DataSet](#) **\$ds** DataSet to fill
- [DataSet](#) \\&**\$ds**

**update DataSet with Control data**

- **Access** public

*DomObject* function TextBox::getDomObject() [[line 188](#)]

**returns a DomObject representing control**

- **Access** public

*bool* function TextBox::onLoad() [[line 315](#)]

**called when control is loaded**

- **Access** protected

`void function TextBox::onRequest(&$request, $request) [line 339]`

**Function Parameters:**

- *object* **\$request** HTTPRequest Object
- *array* **&\$request**

**process the HTTP request array**

- **Access** protected

## Class TimeSelector

[line [22](#)]

**handles date control element creation (string) abstracts away the presentation logic and data access layer the server-side control for WebWidgets**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** bool \$allowNull: specifies whether to allow null values
- **Author** Darnell Shinbine

**TimeSelector::\$allowNull**



*mixed* = false [[line 28](#)]

**specifies whether to allow nulls**

- **Access** protected

*void* function TimeSelector::getDomObject(0) [[line 145](#)]

**Function Parameters:**

- *none* **0**

**returns widget object**

- **Access** public

*void* function TimeSelector::onLoad() [[line 74](#)]

**called when control is loaded**

- **Access** protected

*void* function TimeSelector::onRequest(&\$httpRequest) [[line 88](#)]

**Function Parameters:**

- *array* **&\$httpRequest**

**process the HTTP request array**

- **Access** protected

## Class TreeNode

[line [25](#)]

**Represents a tree node**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$id: Node id
- **Property** string \$textOrHtml: Node text
- **Property** string \$imgSrc: Specifies the path of the icon image
- **Property** bool \$expanded: Specifies if the node is expanded by default
- **Property** [TreeNodeCollection](#) \$childNodes: collection of child nodes
- **Property** [TreeNode](#) \$parentNode: Contains parent node
- **Author** Darnell Shinbine

**TreeNode::\$childNodes**

*TreeNodeCollection* = null [line [55](#)]

**collection of child nodes**

- **Access** protected

### **TreeNode::\$expanded**

*bool* = false [[line 49](#)]

**Specifies if the node is expanded by default**

- **Access** protected

### **TreeNode::\$id**

*string* = " [[line 31](#)]

**node id**

- **Access** protected

### **TreeNode::\$imgSrc**

*string* = " [[line 43](#)]

**specifies the path of the icon image**

- **Access** protected

### **TreeNode::\$parentNode**

*TreeNode* = null [[line 61](#)]

**contains parent node**

- **Access** protected

**TreeNode::\$textOrHtml**

*string* = " [[line 37](#)]

**node text**

- **Access** protected

Constructor *void* function `TreeNode::__construct($id, [$textOrHtml = ''], [$expanded = false], [$imgSrc = null])` [[line 71](#)]

**Function Parameters:**

- *string* **\$id** TreeNode Id
- *string* **\$textOrHtml** TreeNode text
- *bool* **\$expanded** Specifies whether to expand the TreeNode
- *string* **\$imgSrc** Specifies the icon image source

- **Access** public

*bool* function `TreeNode::addChild($treeNode)` [[line 312](#)]

**Function Parameters:**

- [TreeNode](#) **\$treeNode** reference to an instance of a TreeNode object

**adds child TreeNode to current TreeNode**

- **Access** public

*void function* `TreeNode::collapse()` [*line* [459](#)]  
**collapse node**

- **Access** public

*void function* `TreeNode::collapseAll()` [*line* [470](#)]  
**collapses all children**

- **Access** public

*TreeNode function* `TreeNode::createChild([$textOrHTML = ''], [$expanded = false], [$imgSrc = ''])` [*line* [294](#)]  
**Function Parameters:**

- *string* **\$textOrHTML** string value of node
- *bool* **\$expanded** expand node by default
- *string* **\$imgSrc** path to icon image

**creates a child `TreeNode` in the current `TreeNode`**

- **Access** public

*void function* `TreeNode::expand()` [*line* [433](#)]  
**expand node**

- **Access** public

*void* function `TreeNode::expandAll()` [*line* [444](#)]  
**expands all children**

- **Access** public

*TreeNode* function `TreeNode::findChildNode($id)` [*line* [261](#)]  
**Function Parameters:**

- *string* **\$id** id of `TreeNode`

**returns `TreeNode` if node is found in Collection**

- **Access** public

*XMLEntityCollection* function `TreeNode::getAncestry()` [*line* [414](#)]  
**returns node ancestry**

- **Access** public

*TreeNode* function `TreeNode::getChild($id)` [*line* [345](#)]  
**Function Parameters:**

- *string* **\$id** id of node

**returns child node by Id**

- **Access** public

*TreeNodeCollection* function *TreeNode::getChildren()* [[line 364](#)]

**returns array of sibling nodes**

- **Access** public

*TreeNode* function *TreeNode::getParent()* [[line 403](#)]

**get parent node**

- **Access** public

*XMLEntityCollection* function *TreeNode::getSiblings()* [[line 375](#)]

**returns array of sibling nodes**

- **Access** public

*bool* function *TreeNode::hasChild(\$id)* [[line 325](#)]

**Function Parameters:**

- *string* **\$id** id of node

**returns true if child node exists**

- **Access** public

*bool* function TreeNode::hasChildNode(\$id) [*line* [231](#)]

**Function Parameters:**

- *string* **\$id** id of TreeNode

**returns true if node is found in Collection**

- **Access** public

*void* function TreeNode::loadViewState(&\$viewState, \$viewState) [*line* [151](#)]

**Function Parameters:**

- *object* **\$viewState** session array
- *array* **&\$viewState**

**read view state from session**

- **Access** public



`void function TreeNode::requestProcessor($treeViewControllId, &$request, $request) [line 181]`

**Function Parameters:**

- *string* **\$treeViewControllId** TreeView Control Id
- *array* **\$request** Request Parameter Collection
- *array* **&\$request**

## process the HTTP request array

- **Access** public

`void function TreeNode::saveViewState(&$viewState, $viewState) [line 209]`

**Function Parameters:**

- *object* **\$viewState** session array
- *array* **&\$viewState**

## write view state to session

- **Access** public

`void function TreeNode::setParent(&$parentNode, $treeNode) [line 392]`

**Function Parameters:**

- [TreeNode](#) **\$treeNode** reference to an instance of a TreeNode object
- [TreeNode](#) **&\$parentNode**

## set parent node

- **Access** public

## Class `TreeNodeCollection`

*[line [19](#)]*

**Represents a Collection of `TreeNode` objects**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

*bool* function `TreeNodeCollection::add($item)` *[line [52](#)]*

**Function Parameters:**

- [TreeNode](#) `$item`

**add `TreeNode` to Collection**

- **Access** public

*bool* function `TreeNodeCollection::contains($id)` *[line [84](#)]*

**Function Parameters:**

- *string* **\$id** TreeNode id

**returns true if array item is found**

- **Access** public

*int* function TreeNodeCollection::indexOf(\$id) [[line 103](#)]

**Function Parameters:**

- *string* **\$id** TreeNode id

**returns index if value is found in collection**

- **Access** public

*TreeNode* function TreeNodeCollection::itemAt(\$index) [[line 72](#)]

**Function Parameters:**

- *int* **\$index** index of TreeNode

**return TreeNode at a specified index**

- **Access** public

# Class TreeView

[[line 22](#)]

**Represents a TreeView Control**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** bool \$showIcons: Specifies whether to show icons in tree
- **Property** bool \$showRoot: Specifies whether to show the root node
- **Property** rootNode \$rootNode: Contains the root node
- **Author** Darnell Shinbine

**TreeView::\$rootNode**

*TreeNode* = null [[line 40](#)]

**contains the root node**

- **Access** protected

**TreeView::\$showIcons**

*bool* = true [[line 28](#)]

**specifies whether to show icons in tree, Default is true**

- **Access** protected

## TreeView::\$showRoot

*bool* = true [[line 34](#)]

**specifies whether to show the root node**

- **Access** protected

*TreeNode* function TreeView::findTreeItem(\$id) [[line 130](#)]

### **Function Parameters:**

- *string* **\$id** id of node

**traverses all tree nodes and returns true if a node is found**

- **Access** public

*DomObject* function TreeView::getDomObject() [[line 145](#)]

**returns a DomObject representing control**

- **Access** public

*DomObject* function TreeView::getTreeNodeDomObject(&\$treeNode, \$treeNode) [[line 181](#)]

### **Function Parameters:**

- [TreeNode](#) **\$treeNode** TreeNode object
- [TreeNode](#) **&\$treeNode**

returns DomObject for the TreeNode and all child TreeNodes

- **Access** protected

*bool* function TreeView::onLoad() [[line 328](#)]

**called when control is loaded**

- **Access** protected

*void* function TreeView::onLoadViewState(&\$viewState, \$viewState) [[line 314](#)]

**Function Parameters:**

- *object* **\$viewState** session array
- *array* **&\$viewState**

**read view state from session**

- **Access** protected

*void* function TreeView::onRequest(&\$request) [[line 342](#)]

**Function Parameters:**

- *array* **&\$request**

**process the HTTP request array**

- **Access** protected

void function TreeView::onSaveViewState(&\$viewState, \$viewState) [[line 360](#)]

**Function Parameters:**

- *object* **\$viewState** session array
- *array* **&\$viewState**

**write view state to session**

- **Access** protected

## Class View

[[line 24](#)]

**Represents a Page Control**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$contentType: Document content-type
- **Property** string \$charset: Document character set
- **Property** string \$template: Document template
- **Property** array \$parameters: Document parameters
- **Property** [MasterView](#) \$master: Document master page

- **Author** Darnell Shinbine

#### **View::\$charset**

*string* = 'utf-8' [[line 36](#)]

**specifies charset, default is utf-8**

- **Access** protected

#### **View::\$contentType**

*string* = 'text/plain' [[line 30](#)]

**specifies content-Type, default is text/plain**

- **Access** protected

#### **View::\$template**

*string* = " [[line 42](#)]

**specifies view template**

- **Access** protected

Constructor *void* function View::\_\_construct(\$controlId) [[line 81](#)]

#### **Function Parameters:**

- *string* **\$controlId** Control Id



**sets the controlId and prepares the control attributes**

- **Access** public

void function View::add(\$control, &\$control) [[line 165](#)]

**Function Parameters:**

- [WebControlBase](#) **&\$control** instance of a WebControl
- [WebControlBase](#) **\$control**

**adds child control to collection**

- **Access** public

void function View::addHeader(\$header) [[line 288](#)]

**Function Parameters:**

- *string* **\$header** header to add

**add header**

- **Access** public

void function View::assign(\$field, \$value) [[line 178](#)]

**Function Parameters:**

- *string* **\$field** var name
- *mixed* **\$value** var value

## assign template variables to the view

- **Access** public

*string* function View::fetch([\$args = array()]) [*line* [318](#)]

### **Function Parameters:**

- *array* **\$args** widget parameters

## returns an html string

- **Access** public

*void* function View::fetchTemplate() [*line* [441](#)]

### **return template**

- **Access** protected

*DomObject* function View::getDomObject() [*line* [306](#)]

### **returns a DomObject for rendering**

- **Access** public

*void* function View::onInit() [[line 367](#)]

**Event called when control is initiated**

- **Access** protected

*void* function View::onLoad() [[line 381](#)]

**Event called when page and all controls are loaded**

- **Access** protected

*void* function View::onPost(&\$request) [[line 411](#)]

**Function Parameters:**

- *array* **&\$request** request data

**handle post events**

- **Access** protected

*void* function View::onRequest(&\$request) [[line 396](#)]

**Function Parameters:**

- *array* **&\$request** request data

## Event called when request is processed

- **Access** protected

*void function View::render([\$args = array()]) [line [349](#)]*

### **Function Parameters:**

- *array* **\$args** widget parameters

**This method will render the control to the html canvass**

- **Access** public

*void function View::sendHeaders() [line [425](#)]*

### **send headers**

- **Access** protected

*void function View::setData(\$data) [line [269](#)]*

### **Function Parameters:**

- *string* **\$data** raw data

**set data**

- **Access** public

*void function View::setMaster(\$master) [line [239](#)]*

**Function Parameters:**

- [MasterView](#) **\$master** master Page

**assign master view to this view**

- **Access** public

## Class WebControlAttributeCollection

*[line [19](#)]*

**Represents a Collection of WebControl attributes**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

*void function WebControlAttributeCollection::append(\$name, \$value) [line [53](#)]*

**Function Parameters:**

- *string* **\$name** name of attribute
- *string* **\$value** value of attribute

**append attribute of xml node   overwrite attribute with same name**

- **Access** public

## Class WebControlBase

[line [33](#)]

**Provides a web-based server-side UI to manage the HTML that ends up on client machine by abstracting away the presentation logic and data access layer**

- **Package** PHPRum
- **Sub-Package** Web
- **Property** string \$controlId: Control Id
- **Property** bool \$enableViewState: Specifies whether to enable view state preservation
- **Property** bool \$visible: Specifies whether control is visible
- **Property** bool \$escapeOutput: Specifies whether to escape the output
- **Property** string \$ajaxCallback: Specifies the ajax callback uri (optional)
- **Property** string \$ajaxEventHandler: Specifies the name of the ajax event handler javascript function (optional)
- **Property** string \$ajaxHttpRequest: Specifies the name of the ajax HttpRequest object (optional)
- **Property** [WebControlAttributeCollection](#) \$attributes: Collection of attributes
- **Property** [WebControlCollection](#) \$controls: Collection of child controls
- **Property** [WebControlBase](#) \$parent: parent control
- **Property** [DataSet](#) \$dataSource: Reference to data-source

- **Property** mixed \$value: Control value
- **Property** [EventCollection](#) \$events: event collection
- **Author** Darnell Shinbine
- **Abstract Element**

#### **WebControlBase::\$ajaxCallback**

*string* = " [\[line 57\]](#)

**specifies the ajax callback uri (optional)**

- **Access** protected

#### **WebControlBase::\$ajaxEventHandler**

*string* = " [\[line 63\]](#)

**specifies the name of the ajax event handler javascript function (optional)**

- **Access** protected

#### **WebControlBase::\$ajaxHttpRequest**

*string* = " [\[line 69\]](#)

**specifies the name of the ajax HttpRequest object (optional)**

- **Access** protected

### **WebControlBase::\$attributes**

*WebControlAttributeCollection* = null [[line 75](#)]

**collection of arbitrary attributes (for rendering only)**

- **Access** protected

### **WebControlBase::\$controls**

*WebControlCollection* = null [[line 81](#)]

**collection of child controls**

- **Access** protected

### **WebControlBase::\$dataSource**

*DataSet* = null [[line 87](#)]

**instance of the datasource**

- **Access** protected

### **WebControlBase::\$enableViewState**

*bool* = true [[line 39](#)]

**specifies whether the server control persists its view state, Default is true**

- **Access** protected



### WebControlBase::\$escapeOutput

*bool* = true [[line 51](#)]

**determines whether to escape the output**

- **Access** protected

### WebControlBase::\$events

*EventCollection* = null [[line 93](#)]

**event collection**

- **Access** protected

### WebControlBase::\$visible

*bool* = true [[line 45](#)]

**determines whether the control is visible, Default is true**

- **Access** protected

Constructor *void* function WebControlBase::\_\_construct(\$controlId) [[line 120](#)]

#### **Function Parameters:**

- *string* **\$controlId** Control Id

**sets the controlId and prepares the control attributes**

- **Access** public

*void function* WebControlBase::addControl(\$control, &\$control) [*line* [375](#)]

**Function Parameters:**

- [WebControlBase](#) **&\$control** instance of a WebControl
- [WebControlBase](#) **\$control**

**add a child control to collection**

- **Access** protected

*void function* WebControlBase::bind(\\\$ds, \$ds) [*line* [330](#)]

**Function Parameters:**

- *ArrayAccess* **\$ds** data source to attach
- [ModelBase](#) **\\\$ds**

**bind a data source to control**

- **Access** public

*DomObject function* WebControlBase::createDomObject(\$name) [*line* [891](#)]

***Function Parameters:***

- **\$name**

**returns the default DomObject**

- **Access** protected

*string* function WebControlBase::escape(\$string) [*line* [873](#)]

***Function Parameters:***

- *type* **\$string** string to escape

**escape**

- **Access** protected

*string* function WebControlBase::fetch([\$args = array()]) [*line* [730](#)]

***Function Parameters:***

- *array* **\$args** widget parameters

**returns an html string**

- **Access** public

*WebControlBase* function *WebControlBase::findControl(\$controlId)* [[line 494](#)]

**Function Parameters:**

- *string* **\$controlId** control id

**returns WebControl if control is found in Collection**

- **Access** public

*WebControlBase* function *WebControlBase::getControl(\$controlId)* [[line 445](#)]

**Function Parameters:**

- *string* **\$controlId** id of control

**returns reference to child control**

- **Access** public

*DomObject* function *WebControlBase::getDomObject()* [[line 776](#)]

**returns a DomObject representing control**

- **Abstract Element**
- **Access** public

*string* function *WebControlBase::getHTMLControlId()* [[line 754](#)]

**creates controlId string for html**

- **Access** public

*string* function WebControlBase::getHTMLControlIdString() [*line* [765](#)]  
**creates controlId string for html**

- **Access** public

*WebControlBase* function WebControlBase::getParentByType(\$type) [*line* [526](#)]  
**Function Parameters:**

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

**returns 1st parent object by type**

- **Access** public

*string* function WebControlBase::getParentString() [*line* [551](#)]  
**returns string of parent id's seperated by .**

- **Access** public

*string* function WebControlBase::getQueryString([\$queryString = '']) [*line* [911](#)]  
**Function Parameters:**

- *string* **\$queryString** query string

**creates string of all vars currently in the request object seperated by an '&' used to preserve application state**

- **Access** protected

*string* function WebControlBase::getRequestData() [[line 948](#)]

**creates string of all vars currently in the request object seperated by an '&' used to preserve application state**

- **Access** protected

*void* function WebControlBase::handlePostEvents(&\$request) [[line 664](#)]

**Function Parameters:**

- *array* **&\$request** request data

**handle post events**

- **Access** public

*bool* function WebControlBase::hasControl(\$controlId) [[line 465](#)]

**Function Parameters:**

- *string* **\$controlId** control id

**returns Webtrue if control is found in Collection**

- **Access** public

*void* function WebControlBase::init() [[line 569](#)]  
**called when control is initiated**

- **Access** public

*bool* function WebControlBase::isControl(\$controlId) [[line 433](#)]  
**Function Parameters:**

- *string* **\$controlId** control id

**compares control id, returns true if control id matches**

- **Access** public

*void* function WebControlBase::load(&\$request) [[line 591](#)]  
**Function Parameters:**

- *array* **&\$request** request data

**called when all controls are loaded**

- **Access** public

*void function WebControlBase::loadViewState(&\$viewState) [line [611](#)]*

**Function Parameters:**

- *array* **&\$viewState** session data

**read view state from session**

- **Access** public

*void function WebControlBase::onDataBind() [line [833](#)]*

**Event called when DataBind is called**

- **Access** protected

*void function WebControlBase::onInit() [line [798](#)]*

**Event called when control is initiated**

- **Access** protected

*void function WebControlBase::onLoad(&\$request) [line [807](#)]*

**Function Parameters:**

- *array* **&\$request** request data



## Event called when page and all controls are loaded

- **Access** protected

*void function* WebControlBase::onLoadViewState(&\$viewState) [*line* [790](#)]

### **Function Parameters:**

- *array* **&\$viewState** session data

## Event called when view state is loaded

- **Access** protected

*void function* WebControlBase::onPost(&\$request) [*line* [825](#)]

### **Function Parameters:**

- *array* **&\$request** request data

## handle post events

- **Access** protected

*void function* WebControlBase::onPreRender() [*line* [841](#)]

## Event called when control is ready for rendering

- **Access** protected

*void function WebControlBase::onRequest(&\$request) [line [816](#)]*

***Function Parameters:***

- *array* **&\$request** request data

## **Event called when request is processed**

- **Access** protected

*void function WebControlBase::onSaveViewState(&\$viewState) [line [851](#)]*

***Function Parameters:***

- *array* **&\$viewState** session data

## **Event called when view state is written**

- **Access** protected

*void function WebControlBase::onUpdateAjax() [line [859](#)]*

## **Event called on ajax callback**

- **Access** protected

*void* function WebControlBase::preRender() [[line 680](#)]  
**called before control is ready for rendering**

- **Access** public

*void* function WebControlBase::refreshDataSource() [[line 355](#)]  
**refresh data source**

- **Access** public

*bool* function WebControlBase::removeControl(&\$webcontrol) [[line 394](#)]  
**Function Parameters:**

- [WebControlBase](#) &\$webcontrol instance of a WebControl

**remove a child control**

- **Access** public

*void* function WebControlBase::render([\$args = array()]) [[line 743](#)]  
**Function Parameters:**

- *array* \$args widget parameters

**This method will render the control to the html canvass**

- **Access** public

*bool* function WebControlBase::replaceControl(&\$webcontrol1, &\$webcontrol2, &\$oldWebControl, &\$newWebControl) [*line 414*]

**Function Parameters:**

- [WebControlBase](#) &\$oldWebControl instance of a WebControl
- [WebControlBase](#) &\$newWebControl instance of a WebControl
- [WebControlBase](#) &\$webcontrol1
- [WebControlBase](#) &\$webcontrol2

## replace a child control

- **Access** public

*void* function WebControlBase::requestProcessor(&\$request) [*line 636*]

**Function Parameters:**

- *array* &\$request request data

## process the HTTP request array

- **Access** public

*void* function WebControlBase::saveViewState(&\$viewState) [*line 700*]

### ***Function Parameters:***

- `array &$viewState` session data

### **write view state to session**

- **Access** public

*void* function WebControlBase::setParent(\$control, &\$control) [*line* [311](#)]

### ***Function Parameters:***

- [WebControlBase](#) **&\$control** instance of a WebControl
- [WebControlBase](#) **\$control**

### **set parent control**

- **Access** public

*void* function WebControlBase::updateAjax() [*line* [652](#)]

### **updates the ui control asynchronously**

- **Access** public

# Class WebControlCollection

[[line 19](#)]

**Represents a Collection of WebControl objects**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function WebControlCollection::\_\_construct(&\$parent, \$parent) [[line 35](#)]

**Function Parameters:**

- [WebControlBase](#) **\$parent** instance of a WebControlBase object
- [WebControlBase](#) **&\$parent**

## Constructor

- **Access** public

*void* function WebControlCollection::add(\$item) [[line 78](#)]

**Function Parameters:**

- [WebControlBase](#) **\$item**

## add WebControl to Collection

- **Access** public

*bool* function WebControlCollection::contains(\$controlId) [*line* [155](#)]

**Function Parameters:**

- *string* **\$controlId** control id

**returns true if array item is found**

- **Access** public

*int* function WebControlCollection::indexOf(\$controlId) [*line* [174](#)]

**Function Parameters:**

- *string* **\$controlId** control id

**returns index if value is found in collection**

- **Access** public

*WebControlBase* function WebControlCollection::itemAt(\$index) [*line* [143](#)]

**Function Parameters:**

- *int* **\$index** index of WebControlBase

**return WebControl at a specified index**

- **Access** public

*bool* function WebControlCollection::remove(\$item) [*line* [131](#)]

**Function Parameters:**

- [WebControlBase](#) \$item

**remove item from collection**

- **Access** public

*void* function WebControlCollection::replace(\$controlId, \$item) [*line* [100](#)]

**Function Parameters:**

- *string* \$controlId
- [WebControlBase](#) \$item

**replace a WebControl in a Collection**

- **Access** public

## Class WebControlCreateEvent

[*line* [19](#)]

**Provides event handling**



- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function WebControlCreateEvent::\_\_construct() [[line 26](#)]

### **Constructor**

- **Access** public

## Class WebControlCreateEventHandler

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function WebControlCreateEventHandler::\_\_construct(\$callback) [[line 27](#)]

### **Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

## Class WebControlInitEvent

*[line [19](#)]*

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function WebControlInitEvent::\_\_construct() *[line [26](#)]*

## Constructor

- **Access** public

## Class WebControlInitEventHandler

*[line [19](#)]*

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function WebControlInitEventHandler::\_\_construct(\$callback) [*line* [27](#)]

**Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class WebControlLoadEvent

[*line* [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function WebControlLoadEvent::\_\_construct() [[line 26](#)]

## Constructor

- **Access** public

# Class WebControlLoadEventHandler

[[line 19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shimbine

Constructor *void* function WebControlLoadEventHandler::\_\_construct(\$callback) [[line 27](#)]

## **Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class WebControlPreRenderEvent

[line [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function WebControlPreRenderEvent::\_\_construct() [line [26](#)]

**Constructor**

- **Access** public

# Class WebControlPreRenderEventHandler

[line [19](#)]

**Provides event handling**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function WebControlPreRenderEventHandler::\_\_construct(\$callback) [[line 27](#)]

**Function Parameters:**

- *string* **\$callback** call back

## Constructor

- **Access** public

# Class WebServiceBase

[[line 18](#)]

**This class handles all remote procedure calls for a web service**

- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine
- **Abstract Element**

**WebServiceBase::\$remoteProcedures**

*string* = array() [[line 24](#)]

**Contains an array of RPC method calls**

- **Access** protected

## WebServiceBase::\$reserved\_methods

```
array = array(  
    '__construct',  
    '__set',  
    '__get',  
    'getView',  
    'getRoles',  
    'getCacheld',  
    'configure',  
    'handle') [line 36]
```

### array of reserved methods

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

## WebServiceBase::\$view

*View = [[line 30](#)]*

### contains the View object

- **Access** protected

*void function WebServiceBase::configure() [[line 81](#)]*

### configure the server

- **Access** protected

*View function WebServiceBase::getView(\\&\$request, &\$request) [[line 53](#)]*

### **Function Parameters:**

- [HttpRequest](#) &\$request HttpRequest object
- [HttpRequest](#) \\&\$request

return view component for rendering

- **Access** public

void function WebServiceBase::handle(\\&\$request, &\$request) [*line* [75](#)]

**Function Parameters:**

- [HttpRequest](#) &\$request HttpRequest object
- [HttpRequest](#) \\&\$request

this method will handle the web service request

- **Abstract Element**
- **Access** public

## Class WebServiceMethod

[*line* [18](#)]

This class represents a remote procedure call



- **Package** PHPRum
- **Sub-Package** Web
- **Author** Darnell Shinbine

Constructor *void* function WebServiceMethod::\_\_construct(\$function) [[line 29](#)]

**Function Parameters:**

- *string* **\$function** function callback

## Constructor

- **Access** public

*array* function WebServiceMethod::getFunction() [[line 62](#)]

**return the function callback**

- **Access** public

*array* function WebServiceMethod::getParameters() [[line 72](#)]

**return all parameters**

- **Access** public

*void* function WebServiceMethod::setFunction(\$function) [[line 40](#)]

**Function Parameters:**

- *string* **\$function** function callback

## set function

- **Access** public

*void* function WebServiceMethod::setParameters(\$name, \$type) [*line* [52](#)]

### **Function Parameters:**

- *string* **\$name** parameter name
- *string* **\$type** parameter type

## set parameters

- **Access** public

# domobject.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# textnode.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# xmlentity.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# xmlentitycollection.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# xmlexception.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

# xmlparser.class.php

- **Package** PHPRum
- **Author** Darnell Shinbine
- **Copyright** Copyright (c) 2011
- **Filesource** [Source Code for this file](#)
- **License** see

## Class DomObject

[line [23](#)]

**Represents an XHTML DOM element**

- **Package** PHPRum
- **Sub-Package** XML
- **Property** string \$nodeName: name of XHTML DOM entity
- **Property** string \$nodeValue: value of XHTML DOM entity
- **Property** string \$innerHTML: inner html
- **Property** [XMLEntityCollection](#) \$childNodes: XHTML DOM entity children
- **Author** Darnell Shinbine

**DomObject::\$innerHTML**

*string* = " [line [29](#)]

**innerHTML value**



- **Access** protected

*XMLEntity* function DomObject::getChildById(\$id) [*line* [119](#)]

**Function Parameters:**

- *string* **\$id** id

**returns a reference to a child element by id**

- **Access** public

*string* function DomObject::getEntityValue([\$charset = 'utf-8']) [*line* [100](#)]

**Function Parameters:**

- *string* **\$charset** character set

**return formatted entity value**

- **Access** protected

*string* function DomObject::getXMLString([\$indent = 0]) [*line* [141](#)]

**Function Parameters:**

- **\$indent**

**return as XML string**

- **Access** public

*void* function DomObject::render([\$args = array()]) [[line 130](#)]

**Function Parameters:**

- *array* **\$args** node parameters

## output html node

- **Access** public

# Class TextNode

[[line 18](#)]

**Represents a TextNode XML entity**

- **Package** PHPRum
- **Sub-Package** XML
- **Author** Darnell Shinbine

Constructor *void* function TextNode::\_\_construct([\$text = ""]) [[line 27](#)]

**Function Parameters:**

- *string* **\$text** Node text

## Constructor

- **Access** public

*bool* function `TextNode::addChild($node, &$node)` [*line* [127](#)]

### **Function Parameters:**

- [DomObject](#) **&\$node** instance of a DomObject
- [XMLElement](#) **\$node**

## add a reference to a child element

- **Access** public

*void* function `TextNode::appendAttribute($name, $value)` [*line* [94](#)]

### **Function Parameters:**

- *string* **\$name** name of attribute
- *string* **\$value** value of attribute

## append attribute of xml node overwrite attribute with same name

- **Access** public

*string* function XmlNode::getAttribute(\$name) [[line 105](#)]

**Function Parameters:**

- *string* **\$name** name of attribute

## get attribute of xml node

- **Access** public

*XMLEntity* function XmlNode::getChild(\$index) [[line 138](#)]

**Function Parameters:**

- *int* **\$index** index of element

## returns a reference to a child element by index

- **Access** public

*XMLEntity* function XmlNode::getChildByAttribute(\$attribute, \$value, \$index) [[line 160](#)]

**Function Parameters:**

- *string* **\$index** name of element
- **\$attribute**
- **\$value**

## returns a reference to a child element by element attribute

- **Access** public

*XMLEntity* function *TextNode::getChildByName(\$name)* [*line* [149](#)]

**Function Parameters:**

- *string* **\$name** name of element

**returns a reference to a child element by element name**

- **Access** public

*XMLEntityCollection* function *TextNode::getChildrenByName(\$name)* [*line* [171](#)]

**Function Parameters:**

- *string* **\$name** name of elements

**returns all children by name**

- **Access** public

*string* function *TextNode::getXMLString([\$indent = 0])* [*line* [204](#)]

**Function Parameters:**

- **\$indent**

**return XML string**

- **Access** public

*void* function `TextNode::removeAttribute($name)` [*line* [116](#)]

**Function Parameters:**

- *string* **\$name** name of attribute

## remove an attribute from an xml node

- **Access** public

*bool* function `TextNode::removeChild(&$node)` [*line* [182](#)]

**Function Parameters:**

- [DomObject](#) **&\$node** instance of a DomObject

## remove a reference to a child element

- **Access** public

*bool* function `TextNode::replaceChild(&$node1, &$node2)` [*line* [194](#)]

**Function Parameters:**

- [DomObject](#) **&\$node1** instance of a DomObject
- [DomObject](#) **&\$node2** instance of a DomObject

## replace a reference to a child element

- **Access** public

void function XmlNode::setAttribute(\$name, \$value) [[line 81](#)]

### **Function Parameters:**

- *string* **\$name** name of attribute
- *string* **\$value** value of attribute

## set attribute

- **Access** public

# Class XMLEntity

[[line 27](#)]

## **XMLEntity**

Represents an XML entity

- **Package** PHPRum
- **Sub-Package** XML
- **Property** string \$name: name of XML entity

- **Property** string \$value: value of XML entity
- **Property** string \$cdata: XML cdata
- **Property** XMLEntityAttributeCollection \$attributes: XML entity attributes
- **Property** [XMLEntityCollection](#) \$children: XML entity children
- **Author** Darnell Shinbine

## XMLEntity::\$attributes

*XMLEntityAttributeCollection* = array() [[line 51](#)]

### entity attributes

- **Access** protected

## XMLEntity::\$cdata

*string* = " [[line 45](#)]

### cdata value

- **Access** protected

## XMLEntity::\$children

*XMLEntityCollection* = null [[line 57](#)]

### entity children

- **Access** protected



### XMLEntity::\$forceCloseTag

*bool* = false [[line 63](#)]

#### set to force close tag

- **Access** protected

### XMLEntity::\$name

*string* = " [[line 33](#)]

#### entity name

- **Access** protected

### XMLEntity::\$value

*string* = " [[line 39](#)]

#### entity value

- **Access** protected

Constructor *void* function XMLEntity::\_\_construct([\$name = ""]) [[line 73](#)]

#### **Function Parameters:**

- *string* **\$name** Name of XML entity

### Constructor

- **Access** public

*bool* function XMLEntity::addChild(\$node, &\$node) [*line* [301](#)]

**Function Parameters:**

- [XMLEntity](#) &\$node instance of an XMLEntity
- [XMLEntity](#) \$node

## add a reference to a child element

- **Access** public

*void* function XMLEntity::appendAttribute(\$name, \$value) [*line* [263](#)]

**Function Parameters:**

- *string* \$name name of attribute
- *string* \$value value of attribute

## append attribute of xml node overwrite attribute with same name

- **Access** public

*string* function XMLEntity::fetch([\$args = array()]) [*line* [500](#)]

**Function Parameters:**

- *array* \$args node parameters

**return XML node as string**

- **Access** public

*XMLEntity* function XMLEntity::findChildByName(\$name) [[line 327](#)]

**Function Parameters:**

- *string* **\$name** name of element

**returns a reference to a child element by element name**

- **Access** public

*string* function XMLEntity::getAttribute(\$name) [[line 279](#)]

**Function Parameters:**

- *string* **\$name** name of attribute

**get attribute of xml node**

- **Access** public

*XMLEntity* function XMLEntity::getChild(\$index) [[line 312](#)]

**Function Parameters:**

- *int* **\$index** index of element

**returns a reference to a child element by index**

- **Access** public

*XMLEntity* function *XMLEntity::getChildByAttribute(\$attribute, \$value, \$index)* [[line 360](#)]

**Function Parameters:**

- *string* **\$index** name of element
- **\$attribute**
- **\$value**

**returns a reference to a child element by element attribute**

- **Access** public

*XMLEntity* function *XMLEntity::getChildByName(\$name)* [[line 343](#)]

**Function Parameters:**

- *string* **\$name** name of element

**returns a reference to a child element by element name**

- **Access** public

*XMLEntityCollection* function *XMLEntity::getChildrenByName(\$name)* [*line* [377](#)]

**Function Parameters:**

- *string* **\$name** name of elements

**returns all children by name**

- **Access** public

*string* function *XMLEntity::getEntityValue([\$charset = 'utf-8'])* [*line* [234](#)]

**Function Parameters:**

- *string* **\$charset** character set

**get formatted entity value**

- **Access** protected

*string* function *XMLEntity::getXMLString([\$indent = 0])* [*line* [435](#)]

**Function Parameters:**

- **\$indent**

**return XML string**

- **Access** public

*void* function XMLEntity::removeAttribute(\$name) [*line* [290](#)]

**Function Parameters:**

- *string* **\$name** name of attribute

## remove an attribute from an xml node

- **Access** public

*bool* function XMLEntity::removeChild(&\$node) [*line* [394](#)]

**Function Parameters:**

- [XMLEntity](#) **&\$node** instance of a XMLEntity

## remove a reference to a child element

- **Access** public

*bool* function XMLEntity::replaceChild(&\$node1, &\$node2) [*line* [415](#)]

**Function Parameters:**

- [XMLEntity](#) **&\$node1** instance of a XMLEntity
- [XMLEntity](#) **&\$node2** instance of a XMLEntity

## replace a reference to a child element

- **Access** public

*void* function XMLEntity::setAttribute(\$name, \$value) [*line* [250](#)]

**Function Parameters:**

- *string* **\$name** name of attribute
- *string* **\$value** value of attribute

## set entity attribute

- **Access** public

# Class XMLEntityCollection

[*line* [19](#)]

**Represents a Collection of XMLEntity objects**

- **Package** PHPRum
- **Sub-Package** XML
- **Author** Darnell Shinbine

*bool* function XMLEntityCollection::add(\$item) [*line* [51](#)]

**Function Parameters:**

- [XMLEntity](#) \$item

## add XMLEntity to Collection

- **Access** public

*bool* function XMLEntityCollection::contains(\$entity\_name) [*line* [83](#)]

### **Function Parameters:**

- *string* **\$entity\_name** name of entity

## returns true if array item is found

- **Access** public

*int* function XMLEntityCollection::indexOf(\$entity\_name) [*line* [102](#)]

### **Function Parameters:**

- *string* **\$entity\_name** name of entity

## returns index if value is found in collection

- **Access** public

*XMLEntity* function XMLEntityCollection::itemAt(\$index) [*line* [71](#)]

### **Function Parameters:**



- *int* **\$index** index of ActiveRecord

return XMLEntity at a specified index

- **Access** public

## Class XMLException

[line [19](#)]

Represents an XML Exception

- **Package** PHPRum
- **Sub-Package** XML
- **Author** Darnell Shinbine

## Class XMLParser

[line [21](#)]

Provides a way to parse xml documents into XMLEntity objects

- **Package** PHPRum

- **Sub-Package** XML
- **Property** string \$error: parse error
- **Property** array \$data: xml data
- **Author** Darnell Shinbine

#### XMLParser::\$data

*array* = array() [*line* [33](#)]

#### array of xml data

- **Access** protected

#### XMLParser::\$error

*string* = " [*line* [27](#)]

#### parse error

- **Access** protected

*XMLEntity* function XMLParser::parse(\$xml, \$attrs) [*line* [63](#)]

#### **Function Parameters:**

- *string* **\$xml** raw xml data
- *bool* **\$attrs** specifies whether to return attributes in result

#### parse xml data

## Appendix A - Class Trees

# Package PHPRum

## AppConfiguration

- [AppConfiguration](#)

## ApplicationBase

- [ApplicationBase](#)
  - [ConsoleApplicationBase](#)
    - [Deploy](#)
    - [Get](#)
    - [Make](#)
    - [Migrate](#)
    - [Test](#)
  - [WebApplicationBase](#)

## AppMessage

- [AppMessage](#)

## AppMessageType

- [AppMessageType](#)

## AppState

- [AppState](#)

## Authentication

- [Authentication](#)
  - [BasicAuthentication](#)
  - [FormsAuthentication](#)
  - [WebServiceAuthentication](#)

## AuthenticationStatus

- [AuthenticationStatus](#)

## Build

- [Build](#)

## CacheBase

- [CacheBase](#)
  - [DBCACHE](#)
  - [FileCache](#)
  - [MEMCache](#)

## ColumnCompare

- [ColumnCompare](#)

## ColumnSchema

- [ColumnSchema](#)

## ControllerBase

- [ControllerBase](#)
  - [PageControllerBase](#)

## CredentialBase

- [CredentialBase](#)
  - [TableCredential](#)
  - [UserCredential](#)

## DataAdapter

- [DataAdapter](#)
  - [DirDataAdapter](#)
  - [MSSQLDataAdapter](#)
  - [MySQLDataAdapter](#)
  - [MySQLiDataAdapter](#)
  - [PDODataAdapter](#)
  - [TextDataAdapter](#)

## DatabaseSchema

- [DatabaseSchema](#)

## DataSetType

- [DataSetType](#)

## EventBase

- [EventBase](#)
  - [ApplicationRunEvent](#)
  - [FormAjaxPostEvent](#)
  - [FormPostEvent](#)
  - [GridViewChangeOrderEvent](#)
  - [GridViewColumnAjaxPostEvent](#)
  - [GridViewColumnPostEvent](#)
  - [GridViewFilterEvent](#)
  - [GridViewSortEvent](#)
  - [InputAjaxChangeEvent](#)
  - [InputAjaxPostEvent](#)
  - [InputChangeEvent](#)
  - [InputPostEvent](#)
  - [LoginFormSubmitEvent](#)
  - [MasterViewInitEvent](#)
  - [MasterViewLoadEvent](#)
  - [PageControllerCreatePageEvent](#)
  - [PagePostEvent](#)
  - [PageRequestEvent](#)
  - [ReportDetailsEvent](#)
  - [ReportFooterEvent](#)
  - [ReportGroupFooterEvent](#)
  - [ReportGroupHeaderEvent](#)
  - [ReportHeaderEvent](#)
  - [WebApplicationHandleRequestEvent](#)
  - [WebApplicationLoadStateEvent](#)
  - [WebApplicationSaveStateEvent](#)
  - [WebControlCreateEvent](#)
  - [WebControlInitEvent](#)
  - [WebControlLoadEvent](#)
  - [WebControlPreRenderEvent](#)

## EventHandlerBase

- [EventHandlerBase](#)
  - [ApplicationRunEventHandler](#)

- [FormAjaxPostEventHandler](#)
- [FormPostEventHandler](#)
- [GridViewChangeOrderEventHandler](#)
- [GridViewColumnAjaxPostEventHandler](#)
- [GridViewColumnPostEventHandler](#)
- [GridViewFilterEventHandler](#)
- [GridViewSortEventHandler](#)
- [InputAjaxChangeEventHandler](#)
- [InputAjaxPostEventHandler](#)
- [InputChangeEventHandler](#)
- [InputPostEventHandler](#)
- [LoginFormSubmitEventHandler](#)
- [MasterViewInitEventHandler](#)
- [MasterViewLoadEventHandler](#)
- [PageControllerCreatePageEventHandler](#)
- [PagePostEventHandler](#)
- [PageRequestEventHandler](#)
- [ReportDetailsEventHandler](#)
- [ReportFooterEventHandler](#)
- [ReportGroupFooterEventHandler](#)
- [ReportGroupHeaderEventHandler](#)
- [ReportHeaderEventHandler](#)
- [WebApplicationHandleRequestEventHandler](#)
- [WebApplicationLoadStateEventHandler](#)
- [WebApplicationSaveStateEventHandler](#)
- [WebControlCreateEventHandler](#)
- [WebControlInitEventHandler](#)
- [WebControlLoadEventHandler](#)
- [WebControlPreRenderEventHandler](#)

## ExceptionBase

- [ExceptionBase](#)
  - [ActiveRecordException](#)
  - [ArgumentOutOfRangeException](#)
  - [BadMemberCallException](#)
  - [BadMethodCallException](#)
  - [CacheException](#)
  - [DataAdapterException](#)
  - [DatabaseException](#)
  - [IndexOutOfRangeException](#)
  - [InvalidArgumentException](#)
  - [InvalidOperationException](#)

- [IOException](#)
  - [DirectoryNotFoundException](#)
  - [DirectoryNotWritableException](#)
  - [FileLoadException](#)
  - [FileNotFoundException](#)
  - [FileNotWritableException](#)
- [LoggerException](#)
- [MethodNotImplementedException](#)
- [MissingArgumentException](#)
- [QueryException](#)
- [SchemaException](#)
- [TransactionException](#)
- [TranslatorException](#)
- [TypeMismatchException](#)
- [XMLException](#)

## File

- [File](#)

## FileSystem

- [FileSystem](#)

## Folder

- [Folder](#)

## ForwardMethodType

- [ForwardMethodType](#)



## GridViewColumn

- [GridViewColumn](#)
  - [GridViewControlBase](#)
    - [GridViewButton](#)
    - [GridViewCheckBox](#)
    - [GridViewDropDownMenu](#)
    - [GridViewTextBox](#)
  - [GridViewImage](#)
  - [GridViewLink](#)

## HTMLTestReporter

- HtmlReporter
  - [HTMLTestReporter](#)

## HttpRequest

- [HttpRequest](#)

## HttpResponse

- [HttpResponse](#)

## HTTPWebRequest

- [HTTPWebRequest](#)

## HTTPWebResponse

- [HTTPWebResponse](#)

## IMailClient

- [IMailClient](#)

## IteratorBase

- [IteratorBase](#)
  - [CollectionBase](#)
    - [ActiveRecordCollection](#)
    - [AppMessageCollection](#)
    - [EventArgs](#)
    - [EventCollection](#)
    - [EventHandlerCollection](#)
    - [GridViewColumnCollection](#)
    - [KeyCollection](#)
    - [StringCollection](#)
    - [TreeNodeCollection](#)
    - [ValidatorCollection](#)
    - [ValueCollection](#)
    - [WebControlCollection](#)
    - [XMLEntityCollection](#)
  - [DictionaryBase](#)
    - [StringDictionary](#)
      - [AppSettingsCollection](#)
      - [ListItemCollection](#)
      - [WebControlAttributeCollection](#)

## LoggerBase

- [LoggerBase](#)
  - [DBLogger](#)
  - [FileLogger](#)

## MailMessage

- [MailMessage](#)

## MakeBase

- [MakeBase](#)
  - [Controller](#)
  - [FormController](#)
  - [GridViewController](#)
  - [JSONWebService](#)
  - [ListController](#)
  - [Model](#)
  - [PageController](#)
  - [RESTWebService](#)
  - [SOAPWebService](#)

## MigrationBase

- [MigrationBase](#)

## MigrationCompare

- [MigrationCompare](#)

## Migrations

- [Migrations](#)

## ModelBase

- [ModelBase](#)
  - [DataSet](#)
  - [FormModelBase](#)
    - [ActiveRecordBase](#)

## PHPMailClient

- [PHPMailClient](#)

## QueryBuilderBase

- [QueryBuilderBase](#)
  - [MSSQLQueryBuilder](#)
  - [MySQLiQueryBuilder](#)
  - [MySQLQueryBuilder](#)
  - [PDOQueryBuilder](#)

## RelationshipType

- [RelationshipType](#)

## ReportGroup

- [ReportGroup](#)

## Roles

- [Roles](#)

## Rum

- [Rum](#)

## SendMailClient

- [SendMailClient](#)

## Session

- [Session](#)

## SMTPClient

- [SMTPClient](#)

## SSHClientBase

- [SSHClientBase](#)
  - [DeploymentBase](#)

## StreamBase

- [StreamBase](#)
  - [FileStream](#)
  - [SocketStream](#)

## TableSchema

- [TableSchema](#)

## TestCaseBase

- [TestCaseBase](#)
  - [ControllerTestCaseBase](#)
  - [MigrationsTestCaseBase](#)
  - [UnitTestCaseBase](#)

## Tester

- [Tester](#)

## TestReporter

- TextReporter
  - [TestReporter](#)

## Timer

- [Timer](#)

## TransactionBase

- [TransactionBase](#)
  - [MSSQLTransaction](#)
  - [MySQLiTransaction](#)
  - [MySQLTransaction](#)
  - [PDOTransaction](#)

## TranslatorBase

- [TranslatorBase](#)
  - [DBTranslator](#)
  - [FileTranslator](#)

## TreeNode

- [TreeNode](#)

## ValidatorBase

- [ValidatorBase](#)
  - [BooleanValidator](#)
  - [DateTimeValidator](#)
  - [EnumValidator](#)
  - [FileSizeValidator](#)
  - [FileTypeValidator](#)
  - [IntegerValidator](#)
  - [LengthValidator](#)
  - [MatchValidator](#)
  - [NumericValidator](#)
  - [PatternValidator](#)

- [EmailValidator](#)
- [URLValidator](#)
- [RangeValidator](#)
- [RequiredValidator](#)
- [UniqueValidator](#)

## WebControlBase

- [WebControlBase](#)
  - [Fieldset](#)
  - [Form](#)
    - [LoginForm](#)
  - [GridView](#)
  - [InputBase](#)
    - [Button](#)
    - [CheckBox](#)
    - [DateSelector](#)
    - [DateTimeSelector](#)
    - [FileBrowser](#)
    - [HyperLink](#)
    - [ListControlBase](#)
      - [ListBox](#)
        - [DropDownList](#)
      - [RadioButtonList](#)
        - [CheckBoxList](#)
    - [RadioButton](#)
    - [RadioGroup](#)
    - [TextBox](#)
      - [TextArea](#)
    - [TimeSelector](#)
  - [Label](#)
  - [ReportView](#)
  - [TreeView](#)
  - [View](#)



- [MasterView](#)
- [Page](#)

## WebServiceBase

- [WebServiceBase](#)
  - [JSONServiceBase](#)
  - [RESTWebServiceBase](#)
  - [SOAPWebServiceBase](#)

## WebServiceMethod

- [WebServiceMethod](#)

## XMLEntity

- [XMLEntity](#)
  - [DomObject](#)
  - [TextNode](#)

## XMLParser

- [XMLParser](#)

# Index

## A

<a href="#">ActiveRecordCollection::itemAt()</a>	0
<i>return ActiveRecord at a specified index</i>	
<a href="#">ActiveRecordException</a>	0
<i>Represents a Bad Member Call Exception</i>	
<a href="#">ActiveRecordCollection::add()</a>	0
<i>Add an ActiveRecord to the Collection</i>	
<a href="#">ActiveRecordCollection</a>	0
<i>Represents a collection of ActiveRecord</i>	
<a href="#">ActiveRecordBase::toArray()</a>	0
<i>returns an array containing all row data</i>	
<a href="#">applicationrunevent.class.php</a>	0
<a href="#">applicationruneventhandler.class.php</a>	0
<a href="#">appsettingscollection.class.php</a>	0
<a href="#">appmessagecollection.class.php</a>	0
<a href="#">ApplicationRunEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">ApplicationRunEvent</a>	0
<i>Provides event handling</i>	
<a href="#">ActiveRecordBase::save()</a>	0
<i>update DataAdapter</i>	
<a href="#">ActiveRecordBase::more()</a>	0
<i>static method to return an extended DataSet of this type</i>	
<a href="#">ActiveRecordBase::delete()</a>	0
<i>delete ActiveRecordBase</i>	
<a href="#">ActiveRecordBase::deleteAll()</a>	0
<i>delete ActiveRecordBase and all associated ActiveRecordBases (cascading delete)</i>	
<a href="#">ActiveRecordBase::create()</a>	0
<i>static method to create new ActiveRecordBase of this type</i>	
<a href="#">ActiveRecordBase::count()</a>	0
<i>static method to return a count of records of this type</i>	
<a href="#">ActiveRecordBase::beforeUpdate()</a>	0
<i>event called before every update</i>	
<a href="#">ActiveRecordBase::find()</a>	0
<i>static method to find ActiveRecordBase of this type</i>	
<a href="#">ActiveRecordBase::findAll()</a>	0
<i>static method to find ActiveRecordBases of this type</i>	
<a href="#">ActiveRecordBase::last()</a>	0
<i>static method to find the last ActiveRecordBase of this type</i>	
<a href="#">ActiveRecordBase::form()</a>	0
<i>static method to return a Form object</i>	
<a href="#">ActiveRecordBase::first()</a>	0
<i>static method to find the first ActiveRecordBase of this type</i>	
<a href="#">ActiveRecordBase::findById()</a>	0
<i>static method to find ActiveRecordBase by primary key</i>	

<a href="#">AppMessageCollection</a>	0
<i>Represents a collection of Application Messages</i>	
<a href="#">AppMessageCollection::add()</a>	0
<i>add AppMessage to Collection</i>	
<a href="#">Authentication::isProtected()</a>	0
<i>returns true if controller is protected (requires authentication to process)</i>	
<a href="#">Authentication::redirectToLogin()</a>	0
<i>redirect user to login</i>	
<a href="#">Authentication::getAuthMethod()</a>	0
<i>returns authentication method</i>	
<a href="#">Authentication::generateSalt()</a>	0
<i>generates a salt</i>	
<a href="#">Authentication::generateHash()</a>	0
<i>generates a hash based on the password</i>	
<a href="#">Authentication::signout()</a>	0
<i>perform sign out (does not end session)</i>	
<a href="#">AuthenticationStatus</a>	0
<i>Provides application wide authentication</i>	
<a href="#">AuthenticationStatus::lockedOut()</a>	0
<i>returns true if the account is locked out</i>	
<a href="#">AuthenticationStatus::invalidCredentials()</a>	0
<i>returns true if invalid credentials supplied</i>	
<a href="#">AuthenticationStatus::disabled()</a>	0
<i>returns true if the account is locked out</i>	
<a href="#">AuthenticationStatus::authenticated()</a>	0
<i>returns true if the account is authenticated</i>	
<a href="#">Authentication::authorized()</a>	0
<i>returns true if user authenticated</i>	
<a href="#">Authentication::authenticated()</a>	0
<i>returns true if user authenticated</i>	
<a href="#">AppSettingsCollection</a>	0
<i>Represents a Collection of user-defined settings</i>	
<a href="#">AppMessageCollection::toString()</a>	0
<i>convert Iterator object into string</i>	
<a href="#">AppMessageCollection::itemAt()</a>	0
<i>return AppMessage at a specified index</i>	
<a href="#">AppMessageCollection::getByType()</a>	0
<i>return Collection of messages</i>	
<a href="#">authentication.class.php</a>	0
<a href="#">authenticationstatus.class.php</a>	0
<a href="#">Authentication::authenticate()</a>	0
<i>authenticate user/password based on credentials (does not set cookie)</i>	
<a href="#">Authentication::\$isProtected</a>	0
<i>specifies whether the current controller is protected</i>	
<a href="#">Authentication::\$identity</a>	0
<i>authenticated id</i>	
<a href="#">Authentication</a>	0
<i>Provides application wide authentication</i>	
<a href="#">ActiveRecordBase::beforeSave()</a>	0
<i>event called before every save</i>	
<a href="#">ActiveRecordBase::beforeRetrieve()</a>	0
<i>event called before every retrieve</i>	
<a href="#">ApplicationBase::getTranslator()</a>	0

<i>returns the common translator object (overridden this method in the application class)</i>	0
<a href="#">ApplicationBase::handleError()</a>	0
<i>event triggered by an error in the application, can be overridden to provide error handling.</i>	
<a href="#">ApplicationBase::getPluginHooks()</a>	0
<i>get plugin hooks</i>	
<a href="#">ApplicationBase::getMailClient()</a>	0
<i>returns the common mail client object (overridden this method in the application class)</i>	
<a href="#">ApplicationBase::getLogger()</a>	0
<i>returns the common logger object (overridden this method in the application class)</i>	
<a href="#">ApplicationBase::handleException()</a>	0
<i>event triggered by an uncaught Exception thrown in the application, can be overridden to provide error handling.</i>	
<a href="#">ApplicationBase::loadAppConfig()</a>	0
<i>loads config data from an xml file</i>	
<a href="#">AppMessageType::Fail()</a>	0
<i>Specifies the message type as Fail</i>	
<a href="#">AppMessageType</a>	0
<i>Represents an application message type</i>	
<a href="#">AppMessage</a>	0
<i>Represents a single application message</i>	
<a href="#">ApplicationBase::run()</a>	0
<i>begin processing of the servlet</i>	
<a href="#">ApplicationBase::getInstance()</a>	0
<i>Returns a static instance of the app server. This provides a way of accessing app variables from outside the class.</i>	
<a href="#">ApplicationBase::getCache()</a>	0
<i>returns the common cache object (overridden this method in the application class)</i>	
<a href="#">appstate.class.php</a>	0
<a href="#">argumentoutofrangeexception.class.php</a>	0
<a href="#">appmessage.type.class.php</a>	0
<a href="#">appmessage.class.php</a>	0
<a href="#">applicationbase.class.php</a>	0
<a href="#">alloc()</a>	0
<i>Print memory usage</i>	
<a href="#">AppConfiguration</a>	0
<i>Provides access to the Application Configuration</i>	
<a href="#">ApplicationBase::execute()</a>	0
<i>execute the application</i>	
<a href="#">ApplicationBase::\$events</a>	0
<i>event collection</i>	
<a href="#">ApplicationBase</a>	0
<i>This class represents the base application. This class recieves command line input and delegates responceibilities to sub components.</i>	
<a href="#">AppConfiguration::loadAppConfig()</a>	0
<i>loads config data from an xml file</i>	
<a href="#">AppMessageType::Info()</a>	0
<i>Specifies the message type as Notice</i>	
<a href="#">AppMessageType::Success()</a>	0
<i>Specifies the message type as Success</i>	
<a href="#">ActiveRecordBase::afterInsert()</a>	0
<i>event called after every insert</i>	
<a href="#">ActiveRecordBase::afterRetrieve()</a>	0
<i>event called after every retrieve</i>	

<a href="#">ActiveRecordBase::afterDelete()</a>	0
<i>event called after every delete</i>	
<a href="#">ActiveRecordBase::afterCreate()</a>	0
<i>event called after a new ActiveRecordBase is created</i>	
<a href="#">ActiveRecordBase::\$table</a>	0
<i>Specifies the table name</i>	
<a href="#">ActiveRecordBase::afterSave()</a>	0
<i>event called after every save</i>	
<a href="#">ActiveRecordBase::afterUpdate()</a>	0
<i>event called after every update</i>	
<a href="#">ActiveRecordBase::beforeInsert()</a>	0
<i>event called before every insert</i>	
<a href="#">ActiveRecordBase::beforeDelete()</a>	0
<i>event called before every delete</i>	
<a href="#">ActiveRecordBase::beforeCreate()</a>	0
<i>event called before a new ActiveRecordBase is created</i>	
<a href="#">ActiveRecordBase::all()</a>	0
<i>static method to return a DataSet of this type</i>	
<a href="#">ActiveRecordBase::\$relationships</a>	0
<i>Specifies table relationships</i>	
<a href="#">ActiveRecordBase::\$pkey</a>	0
<i>Specifies the primary key (there can only be one primary key defined)</i>	
<a href="#">AppState::Debug()</a>	0
<i>Specifies the application state as Debug</i>	
<a href="#">AppState</a>	0
<i>Represents an application state</i>	
<a href="#">AppMessageType::__toString()</a>	0
<i>Returns a string representing the type</i>	
<a href="#">AppMessageType::Warning()</a>	0
<i>Specifies the message type as Notice</i>	
<a href="#">AppState::On()</a>	0
<i>Specifies the application state as On</i>	
<a href="#">ArgumentOutOfRangeException</a>	0
<i>Represents Argument Out Of Range Exception</i>	
<a href="#">ActiveRecordBase</a>	0
<i>This class represents a table withing a database or an instance of a single record in a table and provides database abstraction.</i>	
<a href="#">activerecordexception.class.php</a>	0
<a href="#">activerecordcollection.class.php</a>	0
<a href="#">activerecordbase.class.php</a>	0
<a href="#">appconfiguration.class.php</a>	0

## B

<a href="#">BooleanValidator</a>	0
<i>Provides basic validation for web controls</i>	
<a href="#">BooleanValidator::onLoad()</a>	0
<i>on load</i>	
<a href="#">BooleanValidator::validate()</a>	0
<i>sets the controlId and prepares the control attributes</i>	
<a href="#">booleanvalidator.class.php</a>	0
<a href="#">BasicAuthentication::signout()</a>	0

<i>perform sign out</i>	0
<a href="#">BasicAuthentication::isAuthUserSet()</a>	0
<i>returns true if auth is set</i>	
<a href="#">BasicAuthentication::sendAuthHeaders()</a>	0
<i>send auth http headers</i>	
<a href="#">button.class.php</a>	0
<a href="#">Button</a>	0
<i>Represents a Button Control</i>	
<a href="#">Button::getDomObject()</a>	0
<i>getDomObject</i>	
<a href="#">Button::onLoad()</a>	0
<i>onLoad</i>	
<a href="#">Button::onRequest()</a>	0
<i>onRequest</i>	
<a href="#">Button::\$text</a>	0
<i>specifies button text</i>	
<a href="#">Button::\$submitText</a>	0
<i>specifies button text to display on submit</i>	
<a href="#">Button::\$disableOnSubmit</a>	0
<i>specifies whether to disable all other buttons on submit</i>	
<a href="#">Button::\$src</a>	0
<i>specifies button image source</i>	
<a href="#">BasicAuthentication::getAuthUser()</a>	0
<i>gets auth user</i>	
<a href="#">BasicAuthentication::authenticated()</a>	0
<i>returns true if user authenticated</i>	
<a href="#">build.class.php</a>	0
<a href="#">Build</a>	0
<i>Provides access to read/write build files</i>	
<a href="#">Build::\$verbose</a>	0
<i>verbose</i>	
<a href="#">BadMethodCallException</a>	0
<i>Represents a Bad Method Call Exception</i>	
<a href="#">BadMemberCallException</a>	0
<i>Represents a Bad Member Call Exception</i>	
<a href="#">badmethodcallexception.class.php</a>	0
<a href="#">breakpoint()</a>	0
<i>Set breakpoint</i>	
<a href="#">Build::build()</a>	0
<i>Rebuild build folder</i>	
<a href="#">Build::clean()</a>	0
<i>Cleans and prepares build folder</i>	
<a href="#">basicauthentication.class.php</a>	0
<a href="#">BasicAuthentication</a>	0
<i>Provides application wide authentication using Basic HTTP Headers</i>	
<a href="#">Build::put()</a>	0
<i>Stores a value identified by a key into cache.</i>	
<a href="#">Build::getFileArray()</a>	0
<i>get id array</i>	
<a href="#">Build::clear()</a>	0
<i>Deletes a value with the specified key from cache</i>	
<a href="#">Build::get()</a>	0
<i>Retrieves a value from cache with a specified key.</i>	

[badmembercallexception.class.php](#) . . . . . 0

## C

[constructor DateTimeSelector::\\_\\_construct\(\)](#) . . . . . 0

*Constructor*

[constructor DateSelector::\\_\\_construct\(\)](#) . . . . . 0

*Constructor*

[constructor Fieldset::\\_\\_construct\(\)](#) . . . . . 0

*Constructor*

[constructor FileBrowser::\\_\\_construct\(\)](#) . . . . . 0

*Constructor*

[constructor Form::\\_\\_construct\(\)](#) . . . . . 0

*Constructor*

[ControllerBase::onLoad\(\)](#) . . . . . 0

*Event called when object created*

[ControllerBase::getView\(\)](#) . . . . . 0

*return view component for rendering*

[ControllerBase::\\$requireSSL](#) . . . . . 0

*Specifies whether the controller requires an SSL connection*

[ControllerBase::\\$outputCache](#) . . . . . 0

*Specifies how long to cache page output in seconds, 0 disables caching*

[constructor ControllerBase::\\_\\_construct\(\)](#) . . . . . 0

*Constructor*

[ControllerBase::getCacheId\(\)](#) . . . . . 0

*this method will generate and return a cache id*

[ControllerBase::getRoles\(\)](#) . . . . . 0

*this method return an array of roles that have been granted access*

[constructor FormAjaxPostEvent::\\_\\_construct\(\)](#) . . . . . 0

*Constructor*

[constructor FormAjaxPostEventHandler::\\_\\_construct\(\)](#) . . . . . 0

*Constructor*

[constructor GridViewColumnPostEvent::\\_\\_construct\(\)](#) . . . . . 0

*Constructor*

[constructor GridViewColumnAjaxPostEventHandler::\\_\\_construct\(\)](#) . . . . . 0

*Constructor*

[constructor GridViewColumnPostEventHandler::\\_\\_construct\(\)](#) . . . . . 0

*Constructor*

[constructor GridViewControlBase::\\_\\_construct\(\)](#) . . . . . 0

[constructor GridViewDropDownMenu::\\_\\_construct\(\)](#) . . . . . 0

[constructor GridViewColumnAjaxPostEvent::\\_\\_construct\(\)](#) . . . . . 0

*Constructor*

[constructor GridViewColumn::\\_\\_construct\(\)](#) . . . . . 0

[constructor FormPostEvent::\\_\\_construct\(\)](#) . . . . . 0

*Constructor*

[constructor FormModelBase::\\_\\_construct\(\)](#) . . . . . 0

*Constructor*

[constructor FormPostEventHandler::\\_\\_construct\(\)](#) . . . . . 0

*Constructor*

[constructor GridView::\\_\\_construct\(\)](#) . . . . . 0

*sets the controlId and prepares the control attributes*

[constructor GridViewButton::\\_\\_construct\(\)](#) . . . . . 0

<a href="#">ControllerBase::\$events</a>	0
<i>event collection</i>	
<a href="#">ControllerBase::\$denyRoles</a>	0
<i>Specifies the roles that have been denied access this page</i>	
<a href="#">constructor RangeValidator:: __construct()</a>	0
<i>RangeValidator</i>	
<a href="#">constructor PatternValidator:: __construct()</a>	0
<i>PatternValidator</i>	
<a href="#">constructor URLValidator:: __construct()</a>	0
<i>URLValidator</i>	
<a href="#">constructor ValidatorBase:: __construct()</a>	0
<i>ValidatorBase</i>	
<a href="#">constructor ValidatorCollection:: __construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor MatchValidator:: __construct()</a>	0
<i>MatchValidator</i>	
<a href="#">constructor LengthValidator:: __construct()</a>	0
<i>LengthValidator</i>	
<a href="#">constructor EmailValidator:: __construct()</a>	0
<i>EmailAddressValidator</i>	
<a href="#">constructor Timer:: __construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor EnumValidator:: __construct()</a>	0
<i>URLValidator</i>	
<a href="#">constructor FileSizeValidator:: __construct()</a>	0
<i>FileSizeValidator</i>	
<a href="#">constructor FileTypeValidator:: __construct()</a>	0
<i>FileTypeValidator</i>	
<a href="#">controllerbase.class.php</a>	0
<a href="#">checkbox.class.php</a>	0
<a href="#">CheckBoxList::getDomObject()</a>	0
<i>returns a DomObject representing control</i>	
<a href="#">CheckBoxList::\$multiple</a>	0
<i>size of listbox</i>	
<a href="#">ControllerBase</a>	0
<i>This class handles all requests for a specific uri.</i>	
<a href="#">ControllerBase::\$allowRoles</a>	0
<i>Specifies the roles that have been granted access this page</i>	
<a href="#">ControllerBase::\$controllerId</a>	0
<i>Specifies the controller id</i>	
<a href="#">CheckBoxList</a>	0
<i>Represents a CheckBoxList Control</i>	
<a href="#">CheckBox::onRequest()</a>	0
<i>process the HTTP request array</i>	
<a href="#">constructor Button:: __construct()</a>	0
<i>Constructor</i>	
<a href="#">checkboxlist.class.php</a>	0
<a href="#">CheckBox</a>	0
<i>Represents a CheckBox Control</i>	
<a href="#">CheckBox::getDomObject()</a>	0
<i>returns a DomObject representing control</i>	
<a href="#">CheckBox::onLoad()</a>	0
<i>called when control is loaded</i>	



<a href="#"><u>constructor GridViewImage:: __construct()</u></a>	0
<a href="#"><u>constructor GridViewLink:: __construct()</u></a>	0
<a href="#"><u>constructor ReportHeaderEvent:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor ReportGroupHeaderEventHandler:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor ReportHeaderEventHandler:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor ReportView:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor TreeNode:: __construct()</u></a>	0
<a href="#"><u>constructor ReportGroupHeaderEvent:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor ReportGroupFooterEventHandler:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor ReportFooterEvent:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor ReportDetailsEventHandler:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor ReportFooterEventHandler:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor ReportGroup:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor ReportGroupFooterEvent:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor View:: __construct()</u></a>	0
sets the controlId and prepares the control attributes	
<a href="#"><u>constructor WebControlBase:: __construct()</u></a>	0
sets the controlId and prepares the control attributes	
<a href="#"><u>constructor WebControlPreRenderEventHandler:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor WebControlPreRenderEvent:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor WebServiceMethod:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor TextNode:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor XMLEntity:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor WebControlLoadEventHandler:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor WebControlLoadEvent:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor WebControlCreateEvent:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor WebControlCollection:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor WebControlCreateEventHandler:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor WebControlInitEvent:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor WebControlInitEventHandler:: __construct()</u></a>	0
Constructor	

<a href="#"><u>constructor ReportDetailsEvent:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor RadioGroup:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor InputPostEvent:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor InputChangeEventHandler:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor InputPostEventHandler:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor Label:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor ListControlBase:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor InputChangeEvent:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor InputBase:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor InputAjaxChangeEvent:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor HyperLink:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor InputAjaxChangeEventHandler:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor InputAjaxPostEvent:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor InputAjaxPostEventHandler:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor LoginForm:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor LoginFormSubmitEvent:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor PagePostEventHandler:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor PagePostEvent:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor PageRequestEvent:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor PageRequestEventHandler:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor RadioButton:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor Page:: __construct()</u></a>	0
sets the controlId and prepares the control attributes	
<a href="#"><u>constructor MasterViewLoadEventHandler:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor MasterView:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor LoginFormSubmitEventHandler:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor MasterViewInitEvent:: __construct()</u></a>	0
Constructor	
<a href="#"><u>constructor MasterViewInitEventHandler:: __construct()</u></a>	0

<a href="#">Constructor</a>	0
<a href="#">constructor MasterViewLoadEvent::__construct()</a>	0
<a href="#">Constructor</a>	0
<a href="#">constructor UnitTestCaseBase::__construct()</a>	0
<a href="#">Constructor</a>	0
<a href="#">ControllerTestCaseBase::tearDown()</a>	0
<a href="#">clean test module</a>	0
<a href="#">constructor DBCache::__construct()</a>	0
<a href="#">Constructor</a>	0
<a href="#">CacheException</a>	0
<a href="#">Represents a Cache Exception</a>	0
<a href="#">constructor MEMCache::__construct()</a>	0
<a href="#">Constructor</a>	0
<a href="#">collectionbase.class.php</a>	0
<a href="#">CollectionBase</a>	0
<a href="#">Provides the base functionality for creating Collections</a>	0
<a href="#">CacheBase::put()</a>	0
<a href="#">Stores a value identified by a key into cache.</a>	0
<a href="#">CacheBase::get()</a>	0
<a href="#">Retrieves a value from cache with a specified key.</a>	0
<a href="#">cacheexception.class.php</a>	0
<a href="#">cachebase.class.php</a>	0
<a href="#">CacheBase</a>	0
<a href="#">Provides access to read/write cache objects</a>	0
<a href="#">CacheBase::clear()</a>	0
<a href="#">Deletes a value with the specified key from cache</a>	0
<a href="#">CacheBase::flush()</a>	0
<a href="#">Deletes all values from cache.</a>	0
<a href="#">constructor CollectionBase::__construct()</a>	0
<a href="#">Constructor</a>	0
<a href="#">CollectionBase::add()</a>	0
<a href="#">add item to Collection</a>	0
<a href="#">constructor IteratorBase::__construct()</a>	0
<a href="#">Constructor</a>	0
<a href="#">constructor EventCollection::__construct()</a>	0
<a href="#">Constructor</a>	0
<a href="#">columncompare.class.php</a>	0
<a href="#">columnschema.class.php</a>	0
<a href="#">ColumnCompare</a>	0
<a href="#">Provides array comparing by type</a>	0
<a href="#">constructor DictionaryBase::__construct()</a>	0
<a href="#">Constructor</a>	0
<a href="#">CollectionBase::removeAt()</a>	0
<a href="#">remove item from array at a specified index</a>	0
<a href="#">CollectionBase::indexOf()</a>	0
<a href="#">returns first index of item found in Collection</a>	0
<a href="#">CollectionBase::contains()</a>	0
<a href="#">returns true if item array contains item</a>	0
<a href="#">CollectionBase::itemAt()</a>	0
<a href="#">return item at a specified index</a>	0
<a href="#">CollectionBase::remove()</a>	0
<a href="#">remove item from array</a>	0
<a href="#">CollectionBase::removeAll()</a>	0

<i>emptys the collection</i>	
<a href="#">constructor WebApplicationSaveStateEventHandler::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor WebApplicationSaveStateEvent::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor EventBase::__construct()</a>	0
<i>Constructor</i>	
<a href="#">ConsoleApplicationBase</a>	0
<i>Provides the base funcionality of a console application. This class recieves command line data and executes the application</i>	
<a href="#">constructor EventHandlerBase::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor GridViewChangeOrderEvent::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor GridViewChangeOrderEventHandler::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor AppMessage::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor ApplicationBase::__construct()</a>	0
<i>Constructor</i>	
<a href="#">callstack()</a>	0
<i>Return callstack string as html</i>	
<a href="#">class search()</a>	0
<i>class_search</i>	
<a href="#">clr()</a>	0
<i>Clear all output buffers</i>	
<a href="#">consoleapplicationbase.class.php</a>	0
<a href="#">constructor AppConfiguration::__construct()</a>	0
<i>Creates an instance of the controller and sets default action map.</i>	
<a href="#">constructor GridViewFilterEvent::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor GridViewFilterEventHandler::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor WebApplicationHandleRequestEvent::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor PageControllerCreatePageEventHandler::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor WebApplicationHandleRequestEventHandler::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor WebApplicationLoadStateEvent::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor WebApplicationLoadStateEventHandler::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor PageControllerCreatePageEvent::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor ApplicationRunEventHandler::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor GridViewSortEventHandler::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor GridViewSortEvent::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor WebApplicationBase::__construct()</a>	0
<i>Constructor</i>	

<a href="#">constructor ActiveRecordBase::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor ApplicationRunEvent::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor ColumnCompare::__construct()</a>	0
<i>Constructor</i>	
<a href="#">ColumnCompare::compareDateString()</a>	0
<i>method compares ints and returns -1, 0, 1</i>	
<a href="#">CredentialBase::authenticate()</a>	0
<i>authenticates password based on the credential</i>	
<a href="#">constructor CredentialBase::__construct()</a>	0
<i>Constructor</i>	
<a href="#">CredentialBase::comparePassword()</a>	0
<i>compare passwords, return true on success</i>	
<a href="#">controllertestcasebase.class.php</a>	0
<a href="#">ControllerTestCaseBase</a>	0
<i>Provides base functionality for the ControllerTestCase</i>	
<a href="#">CredentialBase::\$credential</a>	0
<i>credential data</i>	
<a href="#">CredentialBase</a>	0
<i>Provides application wide authentication</i>	
<a href="#">Controller::make()</a>	0
<i>make</i>	
<a href="#">Controller::ControllerNamespace</a>	0
<i>Default namespace</i>	
<a href="#">constructor MigrationBase::__construct()</a>	0
<i>Constructor</i>	
<a href="#">credentialbase.class.php</a>	0
<a href="#">constructor AuthenticationStatus::__construct()</a>	0
<i>???</i>	
<a href="#">ControllerTestCaseBase::\$controller</a>	0
<i>instance of the controller</i>	
<a href="#">ControllerTestCaseBase::\$view</a>	0
<i>View object</i>	
<a href="#">ControllerTestCaseBase::response()</a>	0
<i>returns the response</i>	
<a href="#">ControllerTestCaseBase::post()</a>	0
<i>simulate a post request</i>	
<a href="#">ControllerTestCaseBase::responseAsXMLEntity()</a>	0
<i>returns the response as XMLEntity object</i>	
<a href="#">ControllerTestCaseBase::setUp()</a>	0
<i>setup test module</i>	
<a href="#">ControllerTestCaseBase::submit()</a>	0
<i>simulate a form submission</i>	
<a href="#">ControllerTestCaseBase::get()</a>	0
<i>simulate a get request</i>	
<a href="#">ControllerTestCaseBase::assertResponse()</a>	0
<i>asserts true if response exists</i>	
<a href="#">ControllerTestCaseBase::assertAssigned()</a>	0
<i>asserts true if var is assigned</i>	
<a href="#">constructor ControllerTestCaseBase::__construct()</a>	0
<i>Constructor</i>	
<a href="#">ControllerTestCaseBase::assertMessage()</a>	0

<i>asserts true if message exists</i>	0
<a href="#">ControllerTestCaseBase::assertNoResponse()</a>	0
<i>asserts true if no response exists that matches \$string</i>	
<a href="#">ControllerTestCaseBase::assertRedirectedTo()</a>	0
<i>asserts true if controller redirected</i>	
<a href="#">Controller</a>	0
<i>Provides functionality to generate controller/view files</i>	
<a href="#">controller.php</a>	0
<a href="#">constructor DatabaseSchema::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor DataAdapter::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor QueryBuilderBase::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor TableSchema::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor TransactionBase::__construct()</a>	0
<i>Constructor</i>	
<a href="#">ColumnSchema::__get()</a>	0
<i>returns an object property</i>	
<a href="#">constructor ColumnSchema::__construct()</a>	0
<i>Constructor</i>	
<a href="#">ColumnCompare::compareNumeric()</a>	0
<i>method compares numbers and returns -1, 0, 1</i>	
<a href="#">ColumnCompare::compareInt()</a>	0
<i>method compares ints and returns -1, 0, 1</i>	
<a href="#">ColumnCompare::compareString()</a>	0
<i>method compares strings and returns -1, 0, 1</i>	
<a href="#">ColumnCompare::compareStringi()</a>	0
<i>method compares strings and returns -1, 0, 1</i>	
<a href="#">ColumnSchema</a>	0
<i>Represents a column schema</i>	
<a href="#">constructor DeploymentBase::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor DBTranslator::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor FileLogger::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor DBLogger::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor MailMessage::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor SendMailClient::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor SMTPClient::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor Session::__construct()</a>	0
<i>Read values from request, clean up variables, and merge get and post requests.</i>	
<a href="#">constructor HTTPResponse::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor TranslatorBase::__construct()</a>	0
<i>Constructor</i>	
<a href="#">constructor FileTranslator::__construct()</a>	0

Constructor	
<a href="#">constructor File::__construct()</a>	0
Constructor	
<a href="#">constructor Folder::__construct()</a>	0
Constructor	
<a href="#">constructor HTTPRequest::__construct()</a>	0
Constructor	
<a href="#">classloader.inc.php</a>	0

## D

<a href="#">DirDataAdapter::insert()</a>	0
<i>attempt to insert a record into the datasource</i>	
<a href="#">DirDataAdapter::open()</a>	0
<i>opens a connection to a filesystem resource</i>	
<a href="#">DirDataAdapter::opened()</a>	0
<i>returns true if a connection to a datasource is currently open</i>	
<a href="#">DirDataAdapter::query()</a>	0
<i>Executes a query procedure on the current connection and return the result</i>	
<a href="#">DirDataAdapter::getLastInsertId()</a>	0
<i>return id of last record inserted</i>	
<a href="#">DirDataAdapter::getAffectedRows()</a>	0
<i>return affected rows</i>	
<a href="#">DirDataAdapter::dropTableSchema()</a>	0
<i>drops a TableSchema object</i>	
<a href="#">DirDataAdapter::escapeString()</a>	0
<i>Returns escaped string</i>	
<a href="#">DirDataAdapter::fill()</a>	0
<i>fetches DataSet from datasource string using source string</i>	
<a href="#">DirDataAdapter::queryBuilder()</a>	0
<i>creates a QueryBuilder object</i>	
<a href="#">DirDataAdapter::update()</a>	0
<i>attempt to update a record in the datasource</i>	
<a href="#">Deploy::handleError()</a>	0
<i>event triggered by an error in the application, can be overridden to provide error handling.</i>	
<a href="#">Deploy::handleException()</a>	0
<i>event triggered by an uncaught Exception thrown in the application, can be overridden to provide error handling.</i>	
<a href="#">DeploymentBase</a>	0
<i>Provides base functionality for deploying applications to remote servers.</i>	
<a href="#">DeploymentBase::\$home_path</a>	0
<i>Specifies the home path on the remote server</i>	
<a href="#">Deploy::getOptions()</a>	0
<i>retrieve command line options</i>	
<a href="#">Deploy::execute()</a>	0
<i>execute the application</i>	
<a href="#">deploy.class.php</a>	0
<a href="#">deploymentbase.class.php</a>	0
<a href="#">Deploy</a>	0
<i>deplomyent script</i>	
<a href="#">DirDataAdapter::delete()</a>	0
<i>attempt to delete a record in the datasource</i>	

<a href="#">DirDataAdapter::close()</a>	0
<i>closes an open connection</i>	
<a href="#">DataSet::setTable()</a>	0
<i>set table</i>	
<a href="#">DataSet::sort()</a>	0
<i>sorts DataSet</i>	
<a href="#">DataSet::toArray()</a>	0
<i>Returns an array containg all rows</i>	
<a href="#">DataSet::update()</a>	0
<i>Update DataSource</i>	
<a href="#">DataSet::setRows()</a>	0
<i>set records</i>	
<a href="#">DataSet::setFields()</a>	0
<i>set fields</i>	
<a href="#">DataSet::requery()</a>	0
<i>Requery datasource</i>	
<a href="#">DataSet::seek()</a>	0
<i>sets result pointer to record based on search criteria</i>	
<a href="#">DataSet::setFieldMeta()</a>	0
<i>set fields</i>	
<a href="#">DataSet::updateRowData()</a>	0
<i>update accessor with data from current record</i>	
<a href="#">DataSet:: _get()</a>	0
<i>returns an object property</i>	
<a href="#">DirDataAdapter::addTableSchema()</a>	0
<i>creats a TableSchema object</i>	
<a href="#">DirDataAdapter::alterTableSchema()</a>	0
<i>alters a TableSchema object</i>	
<a href="#">DirDataAdapter::beginTransaction()</a>	0
<i>creates a Transaction object</i>	
<a href="#">DirDataAdapter::buildSchema()</a>	0
<i>builds a DataBaseSchema object</i>	
<a href="#">DirDataAdapter</a>	0
<i>Represents an open connection to the file system</i>	
<a href="#">DataSetType::OpenStatic()</a>	0
<i>Specifies the DataSet type is OpenStatic</i>	
<a href="#">DataSetType</a>	0
<i>Represents a DataSet type</i>	
<a href="#">DataSetType::OpenDynamic()</a>	0
<i>Specifies the DataSet type is OpenDynamic</i>	
<a href="#">DataSetType::OpenReadonly()</a>	0
<i>Specifies the DataSet type is OpenReadonly</i>	
<a href="#">DeploymentBase::\$release_path</a>	0
<i>Specifies the release path on the remote server</i>	
<a href="#">dbtranslator.class.php</a>	0
<a href="#">DateTimeSelector::\$yearMax</a>	0
<i>specifies the year range</i>	
<a href="#">DateTimeSelector::\$yearMin</a>	0
<i>specifies the year range</i>	
<a href="#">DateTimeSelector::getDomObject()</a>	0
<i>returns widget object</i>	
<a href="#">DateTimeSelector::onLoad()</a>	0
<i>called when control is loaded</i>	



<a href="#">DateTimeSelector::\$allowNull</a>	0
<i>specifies whether to allow nulls</i>	
<a href="#">DateTimeSelector</a>	0
<i>handles date control element creation (string)</i>	
<i>abstracts away the presentation logic and data access layer</i>	
<i>the server-side control for WebWidgets</i>	
<a href="#">DateSelector::getDomObject()</a>	0
<i>returns widget object</i>	
<a href="#">DateSelector::onLoad()</a>	0
<i>called when control is loaded</i>	
<a href="#">DateSelector::onRequest()</a>	0
<i>process the HTTP request array</i>	
<a href="#">DateTimeSelector::onRequest()</a>	0
<i>process the HTTP request array</i>	
<a href="#">DropDownList</a>	0
<i>Represents a DropDownList Control</i>	
<a href="#">DomObject::getChildById()</a>	0
<i>returns a reference to a child element by id</i>	
<a href="#">DomObject::getEntityValue()</a>	0
<i>return formatted entity value</i>	
<a href="#">DomObject::getXMLString()</a>	0
<i>return as XML string</i>	
<a href="#">DomObject::render()</a>	0
<i>output html node</i>	
<a href="#">DomObject::\$innerHtml</a>	0
<i>innerHTML value</i>	
<a href="#">DomObject</a>	0
<i>Represents an XHTML DOM element</i>	
<a href="#">DropDownList::\$listSize</a>	0
<i>size of listbox</i>	
<a href="#">DropDownList::\$multiple</a>	0
<i>size of listbox</i>	
<a href="#">domobject.class.php</a>	0
<a href="#">DateSelector::\$yearMin</a>	0
<i>specifies the year range</i>	
<a href="#">DateSelector::\$yearMax</a>	0
<i>specifies the year range</i>	
<a href="#">DirectoryNotWritableException</a>	0
<i>Represents a Directory Not Writable Exception</i>	
<a href="#">dblogger.class.php</a>	0
<a href="#">DBLogger</a>	0
<i>Provides access to read/write log messages</i>	
<a href="#">DBLogger::flush()</a>	0
<i>This method flushes all messages</i>	
<a href="#">DirectoryNotFoundException</a>	0
<i>Represents a Directory Not Found Exception</i>	
<a href="#">directorynotwritableexception.class.php</a>	0
<a href="#">DBTranslator</a>	0
<i>Represents a message Translator using a Database</i>	
<a href="#">DBTranslator::get()</a>	0
<i>get</i>	
<a href="#">directorynotfoundexception.class.php</a>	0
<a href="#">DBLogger::log()</a>	0

<i>This method writes a log message to memory</i>	0
<a href="#">DBLogger::logs()</a>	0
<i>This method retrieves all log messages</i>	
<a href="#">datetimeselector.class.php</a>	0
<a href="#">dropdownlist.class.php</a>	0
<a href="#">DateSelector</a>	0
<i>handles date control element creation (string)</i>	
<i>abstracts away the presentation logic and data access layer</i>	
<i>the server-side control for WebWidgets</i>	
<a href="#">DateSelector::\$allowNull</a>	0
<i>specifies whether to allow nulls</i>	
<a href="#">dateselector.class.php</a>	0
<a href="#">DateTimeValidator::validate()</a>	0
<i>sets the controlId and prepares the control attributes</i>	
<a href="#">datetimevalidator.class.php</a>	0
<a href="#">DateTimeValidator</a>	0
<i>Provides basic validation for web controls</i>	
<a href="#">DateTimeValidator::onLoad()</a>	0
<i>on load</i>	
<a href="#">DataSet::prev()</a>	0
<i>Moves result pointer to previous record</i>	
<a href="#">DataSet::pageCount()</a>	0
<i>Returns the number of pages with data in a DataSet object</i>	
<a href="#">DataAdapter::\$args</a>	0
<i>A collection of connection variables</i>	
<a href="#">DataAdapter::\$cacheEnabled</a>	0
<i>Specifies whether caching is enabled</i>	
<a href="#">DataAdapter::\$cacheExpires</a>	0
<i>Specifies the caching expiration in seconds</i>	
<a href="#">DataAdapter::\$charset</a>	0
<i>Specifies the character set</i>	
<a href="#">DataAdapter</a>	0
<i>Provides access to a generic data source</i>	
<a href="#">dirdataadapter.class.php</a>	0
<a href="#">databaseschema.class.php</a>	0
<a href="#">dataset.class.php</a>	0
<a href="#">datasettype.class.php</a>	0
<a href="#">DataAdapter::\$collation</a>	0
<i>Specifies the db collation</i>	
<a href="#">DataAdapter::addTableSchema()</a>	0
<i>creates a TableSchema object</i>	
<a href="#">DataAdapter::delete()</a>	0
<i>attempt to delete a record in the datasource</i>	
<a href="#">DataAdapter::disableCaching()</a>	0
<i>disable caching</i>	
<a href="#">DataAdapter::dropTableSchema()</a>	0
<i>drops a TableSchema object</i>	
<a href="#">DataAdapter::enableCaching()</a>	0
<i>enable caching</i>	
<a href="#">DataAdapter::create()</a>	0
<i>creates a connection object to a datasource</i>	
<a href="#">DataAdapter::close()</a>	0
<i>closes an open connection</i>	

<a href="#">DataAdapter::alterTableSchema()</a>	0
<i>alters a TableSchema object</i>	
<a href="#">DataAdapter::beginTransaction()</a>	0
<i>creates a Transaction object</i>	
<a href="#">DataAdapter::buildSchema()</a>	0
<i>builds a DataBaseSchema object</i>	
<a href="#">databaseexception.class.php</a>	0
<a href="#">dataadapterexception.class.php</a>	0
<a href="#">DBCACHE::get()</a>	0
<i>Retrieves a value from cache with a specified key.</i>	
<a href="#">DBCACHE::getIdArray()</a>	0
<i>get id array</i>	
<a href="#">DBCACHE::put()</a>	0
<i>Stores a value identified by a key into cache.</i>	
<a href="#">dictionarybase.class.php</a>	0
<a href="#">DBCACHE::flush()</a>	0
<i>Deletes all values from cache.</i>	
<a href="#">DBCACHE::clear()</a>	0
<i>Deletes a value with the specified key from cache</i>	
<a href="#">dmp()</a>	0
<i>Print variable to screen</i>	
<a href="#">dbcache.class.php</a>	0
<a href="#">DBCACHE</a>	0
<i>Provides access to read/write cache objects from a database</i>	
<a href="#">DictionaryBase</a>	0
<i>Provides the base functionality for creating Dictionaries</i>	
<a href="#">DictionaryBase::add()</a>	0
<i>Adds a new key/item pair to a Dictionary object if key does not already exist</i>	
<a href="#">DictionaryBase::remove()</a>	0
<i>remove key/value pair from Dictionary</i>	
<a href="#">DictionaryBase::removeAll()</a>	0
<i>emptys the dictionary</i>	
<a href="#">DictionaryBase::removeAt()</a>	0
<i>remove key/value of a specified index</i>	
<a href="#">dataadapter.class.php</a>	0
<a href="#">DictionaryBase::keyAt()</a>	0
<i>returns the key of a specified index</i>	
<a href="#">DictionaryBase::itemAt()</a>	0
<i>returns the value of a specified index</i>	
<a href="#">DictionaryBase::contains()</a>	0
<i>returns a Boolean value that indicates whether a specified key exists in the Dictionary object</i>	
<a href="#">DictionaryBase::indexOf()</a>	0
<i>returns index of key found in Dictionary</i>	
<a href="#">DictionaryBase::item()</a>	0
<i>return the value of a key in the Dictionary</i>	
<a href="#">DataAdapter::escapeString()</a>	0
<i>Returns an escaped string</i>	
<a href="#">DataAdapter::execute()</a>	0
<i>Executes a query procedure on the current connection</i>	
<a href="#">DataSet::delete()</a>	0
<i>Delete record from DataSource</i>	
<a href="#">DataSet::eof()</a>	0
<i>Returns true on end of file</i>	

<a href="#">DataSet::filter()</a>	0
filter DataSet by removing all records	
<a href="#">DataSet::first()</a>	0
Moves result pointer to first record	
<a href="#">DataSet::bof()</a>	0
returns true at beggining of file	
<a href="#">DataSet::addNew()</a>	0
create new DataSet	
<a href="#">DatabaseSchema::seek()</a>	0
seek	
<a href="#">DatabaseSchema::__get()</a>	0
returns an object property	
<a href="#">DataSet</a>	0
Represents a DataSet	
<a href="#">DataSet::getAvg()</a>	0
returns average of values in column	
<a href="#">DataSet::getCSVString()</a>	0
Returns a DataSet as a CSV string	
<a href="#">DataSet::last()</a>	0
Moves result pointer to last record	
<a href="#">DataSet::move()</a>	0
Moves record pointer to specified record	
<a href="#">DataSet::next()</a>	0
Moves result pointer to next record	
<a href="#">DataSet::page()</a>	0
Moves record pointer to first record on specified page	
<a href="#">DataSet::insert()</a>	0
Inserts record into DataSource	
<a href="#">DataSet::getXMLString()</a>	0
Returns a DataSet as an XML string	
<a href="#">DataSet::getMax()</a>	0
returns max of values in column	
<a href="#">DataSet::getMin()</a>	0
returns min of values in column	
<a href="#">DataSet::getSum()</a>	0
returns sum of all values in column	
<a href="#">DatabaseSchema</a>	0
Represents a database schema	
<a href="#">DatabaseException</a>	0
Represents a DataBase Exception	
<a href="#">DataAdapter::getLastQueryTime()</a>	0
return last executed query time in milliseconds	
<a href="#">DataAdapter::getQueryCount()</a>	0
returns the query count	
<a href="#">DataAdapter::getQueryTime()</a>	0
returns the total query time	
<a href="#">DataAdapter::getSchema()</a>	0
returns a DataBaseSchema object	
<a href="#">DataAdapter::getLastQuery()</a>	0
return last executed query	
<a href="#">DataAdapter::getLastInsertId()</a>	0
return id of last record inserted	
<a href="#">DataAdapter::executeBatch()</a>	0

<i>Executes a query batch or procedure on the current connection</i>	0
<a href="#">DataAdapter::fill()</a>	0
<i>fetches DataSet from datasource string using source string</i>	
<a href="#">DataAdapter::getAffectedRows()</a>	0
<i>return affected rows</i>	
<a href="#">DataAdapter::insert()</a>	0
<i>attempt to insert a record into the datasource</i>	
<a href="#">DataAdapter::open()</a>	0
<i>opens a connection to a datasource</i>	
<a href="#">DataAdapter::runQuery()</a>	0
<i>Executes a query procedure on the current connection and return the result</i>	
<a href="#">DataAdapter::update()</a>	0
<i>attempt to update a record in the datasource</i>	
<a href="#">DataAdapterException</a>	0
<i>Represents a DataAdapter Exception</i>	
<a href="#">DataAdapter::rebuildSchema()</a>	0
<i>rebuilds the DataBaseSchema object</i>	
<a href="#">DataAdapter::queryBuilder()</a>	0
<i>creates a QueryBuilder object</i>	
<a href="#">DataAdapter::openDataSet()</a>	0
<i>opens a DataSet specified by the source</i>	
<a href="#">DataAdapter::opened()</a>	0
<i>returns true if a connection to a datasource is currently open</i>	
<a href="#">DataAdapter::query()</a>	0
<i>Executes a query procedure on the current connection and return the result</i>	
<a href="#">debug.inc.php</a>	0
<i>debug tools</i>	

## E

<a href="#">EventHandlerCollection::raise()</a>	0
<i>raise all events</i>	
<a href="#">EventHandlerCollection::registerEventHandler()</a>	0
<i>register event handler</i>	
<a href="#">EventHandlerCollection</a>	0
<i>Represents a Collection of EventHandlerBase objects</i>	
<a href="#">EventCollection::registerEventHandler()</a>	0
<i>register event handler</i>	
<a href="#">EventCollection::raise()</a>	0
<i>raise all events</i>	
<a href="#">emailvalidator.class.php</a>	0
<a href="#">enumvalidator.class.php</a>	0
<a href="#">EnumValidator::onLoad()</a>	0
<i>on load</i>	
<a href="#">EnumValidator::validate()</a>	0
<i>sets the controlId and prepares the control attributes</i>	
<a href="#">EnumValidator</a>	0
<i>Provides basic validation for web controls</i>	
<a href="#">EmailValidator::onLoad()</a>	0
<i>on load</i>	
<a href="#">EmailValidator</a>	0
<i>Provides basic validation for web controls</i>	

<a href="#">EventCollection::indexOf()</a>	0
<i>returns first index of item found in Collection</i>	
<a href="#">EventCollection::contains()</a>	0
<i>returns true if item array contains item</i>	
<a href="#">EventHandlerBase</a>	0
<i>Provides basic event handling</i>	
<a href="#">EventHandlerBase::raise()</a>	0
<i>raise event</i>	
<a href="#">EventBase</a>	0
<i>Provides basic event handling</i>	
<a href="#">exceptionbase.class.php</a>	0
<a href="#">eventhandlerbase.class.php</a>	0
<a href="#">ExceptionBase</a>	0
<i>Represents a base Exception</i>	
<a href="#">eventargs.class.php</a>	0
<a href="#">EventCollection</a>	0
<i>Represents a Collection of EventHandlerBase objects</i>	
<a href="#">EventArgs</a>	0
<i>Represents a Collection of Event args</i>	
<a href="#">eventhandlercollection.class.php</a>	0
<a href="#">eventcollection.class.php</a>	0
<a href="#">eventbase.class.php</a>	0

## F

<a href="#">FileBrowser</a>	0
<i>Represents a FileBrowser Control</i>	
<a href="#">FileBrowser::getDomObject()</a>	0
<i>returns a DomObject representing control</i>	
<a href="#">FileBrowser::getFileInfo()</a>	0
<i>returns information on the uploaded file</i>	
<a href="#">Fieldset::validate()</a>	0
<i>validate all controls in form object</i>	
<a href="#">Fieldset::readDataSource()</a>	0
<i>update Control with data from the data source</i>	
<a href="#">Fieldset::getDomObject()</a>	0
<i>returns a DomObject representing control</i>	
<a href="#">Fieldset::onLoad()</a>	0
<i>called when control is loaded</i>	
<a href="#">FileBrowser::getFileRawData()</a>	0
<i>return raw file data</i>	
<a href="#">FileBrowser::onLoad()</a>	0
<i>called when control is loaded</i>	
<a href="#">Form::\$ajaxValidation</a>	0
<i>turn on or off ajax validation</i>	
<a href="#">Form::\$autoFocus</a>	0
<i>turn on or off auto focusing</i>	
<a href="#">Form::\$encodeType</a>	0
<i>Form encoding type</i>	
<a href="#">Form::\$ajaxPostBack</a>	0
<i>turn on or off ajax post backs</i>	
<a href="#">Form::\$action</a>	0

<i>URL to send form data, Default is self</i>	0
<a href="#">FileBrowser::onRequest()</a>	0
<i>process the HTTP request array</i>	
<a href="#">Form</a>	0
<i>Represents a Form</i>	
<a href="#">Fieldset::focus()</a>	0
<i>sets focus to the control</i>	
<a href="#">Fieldset::fillDataSource()</a>	0
<i>update data source with Control data</i>	
<a href="#">formajaxposteventhandler.class.php</a>	0
<a href="#">formpostevent.class.php</a>	0
<a href="#">formposteventhandler.class.php</a>	0
<a href="#">formajaxpostevent.class.php</a>	0
<a href="#">FileTypeValidator::validate()</a>	0
<i>validates the control</i>	
<a href="#">FileTypeValidator::addType()</a>	0
<i>adds a file type to the list of allowed types</i>	
<a href="#">FileTypeValidator::onLoad()</a>	0
<i>on load</i>	
<a href="#">formmodelbase.class.php</a>	0
<a href="#">forwardmethodtype.class.php</a>	0
<a href="#">Fieldset::\$legend</a>	0
<i>Fieldset legend</i>	
<a href="#">Fieldset::add()</a>	0
<i>adds child control to collection</i>	
<a href="#">Fieldset</a>	0
<i>Represents a Fieldset</i>	
<a href="#">form.class.php</a>	0
<a href="#">fieldset.class.php</a>	0
<a href="#">filebrowser.class.php</a>	0
<a href="#">Form::\$forward</a>	0
<i>controller to forward to, Default is the current controller</i>	
<a href="#">Form::\$hiddenField</a>	0
<i>specifies whether to check for hidden honeypot field before processing request</i>	
<a href="#">FormModelBase::\$rule_mappings</a>	0
<i>Contains rule mappings</i>	
<a href="#">FormModelBase::form()</a>	0
<i>static method to return a Form object</i>	
<a href="#">FormModelBase::registerFieldType()</a>	0
<i>register new field type</i>	
<a href="#">FormModelBase::\$rules</a>	0
<i>Contains an associative array of field names mapped to rules</i>	
<a href="#">FormModelBase::\$field_mappings</a>	0
<i>Contains field mappings</i>	
<a href="#">FormModelBase</a>	0
<i>This class represents a form.</i>	
<a href="#">FormModelBase::\$fields</a>	0
<i>Contains an associative array of field names mapped to field types</i>	
<a href="#">FormModelBase::registerRuleType()</a>	0
<i>register new rule type</i>	
<a href="#">FormModelBase::save()</a>	0
<i>save model state</i>	
<a href="#">ForwardMethodType::Request()</a>	0

<i>Specifies the application forward method as Request</i>	0
<a href="#">ForwardMethodType::URI()</a>	0
<i>Specifies the application forward method as URI</i>	
<a href="#">ForwardMethodType</a>	0
<i>Represents an application forward method</i>	
<a href="#">FormPostEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">FormModelBase::toArray()</a>	0
<i>converts the form model into an array</i>	
<a href="#">FormPostEvent</a>	0
<i>Provides event handling</i>	
<a href="#">FormAjaxPostEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">FormAjaxPostEvent</a>	0
<i>Provides event handling</i>	
<a href="#">Form::addParameter()</a>	0
<i>add parameter</i>	
<a href="#">Form::begin()</a>	0
<i>renders form open tag</i>	
<a href="#">Form::end()</a>	0
<i>renders form close tag</i>	
<a href="#">Form::add()</a>	0
<i>adds child control to collection</i>	
<a href="#">Form::\$submitted</a>	0
<i>set if the form was submitted</i>	
<a href="#">Form::\$method</a>	0
<i>method to send form data, Default is 'POST'</i>	
<a href="#">Form::\$parameters</a>	0
<i>array of variables to post on submit</i>	
<a href="#">Form::getDomObject()</a>	0
<i>returns a DomObject representing control</i>	
<a href="#">Form::getFormDomObject()</a>	0
<i>returns form as widget</i>	
<a href="#">Form::save()</a>	0
<i>update DataSet using values from child controls</i>	
<a href="#">Form::validate()</a>	0
<i>validate all controls in form object</i>	
<a href="#">Form::onRequest()</a>	0
<i>process the HTTP request array</i>	
<a href="#">Form::onPost()</a>	0
<i>handle post events</i>	
<a href="#">Form::onDataBind()</a>	0
<i>bind all child controls in form to datasource</i>	
<a href="#">Form::onLoad()</a>	0
<i>called when control is loaded</i>	
<a href="#">FileTypeValidator</a>	0
<i>Provides basic validation for web controls</i>	
<a href="#">FileSizeValidator::validate()</a>	0
<i>validates the control</i>	
<a href="#">File::readLastLine()</a>	0
<i>returns the last line of text file <b>**without**</b> reading the entire file</i>	
<a href="#">File::rename()</a>	0
<i>rename file</i>	



<a href="#">FileLoadException</a>	0
<i>Represents a File Load Exception</i>	
<a href="#">File::move()</a>	0
<i>move file to destination</i>	
<a href="#">File::delete()</a>	0
<i>delete file</i>	
<a href="#">File::changeFileMode()</a>	0
<i>change file mode</i>	
<a href="#">File::copy()</a>	0
<i>copy file to destination</i>	
<a href="#">FileNotFoundException</a>	0
<i>Represents a File Not Found Exception</i>	
<a href="#">FileNotWritableException</a>	0
<i>Represents a File Not Writable Exception</i>	
<a href="#">FileSystem::copy()</a>	0
<i>Recursively copy a file or folder and its contents</i>	
<a href="#">FileSystem::delete()</a>	0
<i>Recursively delete a file or folder and its contents using wild cards</i>	
<a href="#">FileSystem::move()</a>	0
<i>Recursively move a file or folder and its contents</i>	
<a href="#">FileSystem::changeFileMode()</a>	0
<i>Change the file mode (chmod)</i>	
<a href="#">FileSystem</a>	0
<i>Provides access to the file system</i>	
<a href="#">FileStream</a>	0
<i>Provides a way to stream data to and from a file</i>	
<a href="#">FileStream::open()</a>	0
<i>opens a text or binary file for input output</i>	
<a href="#">File</a>	0
<i>Represents a file on the file system</i>	
<a href="#">folder.class.php</a>	0
<a href="#">FileCache::getIdArray()</a>	0
<i>get id array</i>	
<a href="#">FileCache::put()</a>	0
<i>Stores a value identified by a key into cache.</i>	
<a href="#">filetranslator.class.php</a>	0
<a href="#">FileCache::get()</a>	0
<i>Retrieves a value from cache with a specified key.</i>	
<a href="#">FileCache::flush()</a>	0
<i>Deletes all values from cache.</i>	
<a href="#">FileCache</a>	0
<i>Provides access to read/write cache files</i>	
<a href="#">FileCache::clear()</a>	0
<i>Deletes a value with the specified key from cache</i>	
<a href="#">FileTranslator</a>	0
<i>Represents a message Translator using a File</i>	
<a href="#">FileTranslator::get()</a>	0
<i>get</i>	
<a href="#">filestream.class.php</a>	0
<a href="#">filesystem.class.php</a>	0
<a href="#">filenotwritableexception.class.php</a>	0
<a href="#">filenotfoundexception.class.php</a>	0
<a href="#">file.class.php</a>	0

<a href="#">fileloadexception.class.php</a>	0
<a href="#">FileSystem::removeDirectory()</a>	0
<i>Recursively delete a dir and its contents</i>	
<a href="#">FileSystem::rename()</a>	0
<i>Rename a file or folder to another location</i>	
<a href="#">FormsAuthentication::getAuthCookie()</a>	0
<i>gets auth cookie</i>	
<a href="#">FormsAuthentication::getAuthSecret()</a>	0
<i>gets auth secret</i>	
<a href="#">FormsAuthentication::isAuthCookieSet()</a>	0
<i>returns true if auth cookie is set</i>	
<a href="#">FormsAuthentication::authenticateSecret()</a>	0
<i>authenticate secret</i>	
<a href="#">FormsAuthentication::authenticated()</a>	0
<i>returns true if user authenticated</i>	
<a href="#">formsauthentication.class.php</a>	0
<a href="#">FormsAuthentication</a>	0
<i>Provides application wide authentication using Forms</i>	
<a href="#">FormsAuthentication::redirectFromLoginPage()</a>	0
<i>sets auth cookie and redirects to original requested resource</i>	
<a href="#">FormsAuthentication::redirectToLoginPage()</a>	0
<i>redirects to login page</i>	
<a href="#">FileSizeValidator</a>	0
<i>Provides basic validation for web controls</i>	
<a href="#">FileSizeValidator::onLoad()</a>	0
<i>on load</i>	
<a href="#">filetypevalidator.class.php</a>	0
<a href="#">filesizevalidator.class.php</a>	0
<a href="#">FormsAuthentication::setAuthCookie()</a>	0
<i>sets auth cookie</i>	
<a href="#">FormsAuthentication::signout()</a>	0
<i>perform sign out (does not end session)</i>	
<a href="#">FormController::make()</a>	0
<i>make</i>	
<a href="#">FormController::ModelNamespace</a>	0
<i>Default namespace</i>	
<a href="#">Folder::move()</a>	0
<i>move folder to destination</i>	
<a href="#">Folder::rename()</a>	0
<i>rename folder</i>	
<a href="#">filelogger.class.php</a>	0
<a href="#">Folder::delete()</a>	0
<i>delete folder</i>	
<a href="#">Folder::copy()</a>	0
<i>copy folder to destination</i>	
<a href="#">Folder</a>	0
<i>Represents a folder on the file system</i>	
<a href="#">Folder::changeFileMode()</a>	0
<i>change file mode</i>	
<a href="#">FileLogger</a>	0
<i>Provides access to read/write log messages</i>	
<a href="#">FileLogger::flush()</a>	0
<i>This method flushes all messages</i>	

<a href="#">FormController::ControllerNamespace</a>	0
<i>Default namespace</i>	
<a href="#">FormController::DefaultBaseController</a>	0
<i>DefaultBaseController</i>	
<a href="#">FormController</a>	0
<i>Provides functionality to generate controller/view files</i>	
<a href="#">formcontroller.php</a>	0
<a href="#">FileLogger::log()</a>	0
<i>This method writes a log message to memory</i>	
<a href="#">FileLogger::logs()</a>	0
<i>This method retrieves all log messages</i>	
<a href="#">filecache.class.php</a>	0

## G

<a href="#">GridViewColumn::\$ScanFilter</a>	0
<i>specifies whether column can be filtered</i>	
<a href="#">GridViewColumn::\$className</a>	0
<i>class name</i>	
<a href="#">GridViewColumn::\$dataField</a>	0
<i>datefield</i>	
<a href="#">GridViewColumn</a>	0
<i>Represents a GridView column</i>	
<a href="#">GridViewCheckBox::getItemText()</a>	0
<i>get item text</i>	
<a href="#">GridViewButton::onPost()</a>	0
<i>handle post events</i>	
<a href="#">GridViewCheckBox</a>	0
<i>Represents a GridView CheckBox</i>	
<a href="#">GridViewColumn::\$events</a>	0
<i>event collection</i>	
<a href="#">GridViewColumn::\$footerText</a>	0
<i>footer text</i>	
<a href="#">GridViewColumn::\$onmouseout</a>	0
<i>specifies the action to take on onmouseout events</i>	
<a href="#">GridViewColumn::\$onmouseover</a>	0
<i>specifies the action to take on mouseover events</i>	
<a href="#">GridViewColumn::getRequestData()</a>	0
<i>creates string of all vars currently in the request object seperated by an '&amp;'</i>	
<i>used to preserve application state</i>	
<a href="#">GridViewColumn::\$ondblclick</a>	0
<i>specifies the action to take on double click events</i>	
<a href="#">GridViewColumn::\$onclick</a>	0
<i>specifies the action to take on click events</i>	
<a href="#">GridViewColumn::\$headerText</a>	0
<i>header text</i>	
<a href="#">GridViewColumn::\$itemText</a>	0
<i>item text</i>	
<a href="#">GridViewButton::getItemText()</a>	0
<i>get item text</i>	
<a href="#">GridViewButton::\$confirmation</a>	0
<i>confirmation message</i>	

<a href="#">GridView::getRowFooter()</a>	0
<i>generic method for handling the table footer</i>	
<a href="#">GridView::getRowHeader()</a>	0
<i>generic method for handling the row header</i>	
<a href="#">GridView::onDataBind()</a>	0
<i>bind control to data</i>	
<a href="#">GridView::getRowFilter()</a>	0
<i>generic method for handling the row header</i>	
<a href="#">GridView::getRowBody()</a>	0
<i>generic method for handling the row body</i>	
<a href="#">GridView::getDomObject()</a>	0
<i>returns a DomObject representing control</i>	
<a href="#">GridView::getPagination()</a>	0
<i>generic method for handling the table pagination</i>	
<a href="#">GridView::onLoad()</a>	0
<i>called when control is loaded</i>	
<a href="#">GridView::onLoadViewState()</a>	0
<i>read view state from session</i>	
<a href="#">GridView::setFilterValues()</a>	0
<i>set filter values</i>	
<a href="#">GridViewButton</a>	0
<i>Represents a GridView button</i>	
<a href="#">GridViewButton::\$canFilter</a>	0
<i>specifies whether column can be filtered</i>	
<a href="#">GridView::onUpdateAjax()</a>	0
<i>Event called on ajax callback</i>	
<a href="#">GridView::onSaveViewState()</a>	0
<i>write view state to session</i>	
<a href="#">GridView::onPost()</a>	0
<i>handle post events</i>	
<a href="#">GridView::onRequest()</a>	0
<i>process the HTTP request array</i>	
<a href="#">GridViewColumn::onLoad()</a>	0
<i>handle load events</i>	
<a href="#">GridViewColumn::onPost()</a>	0
<i>handle post events</i>	
<a href="#">GridViewControlBase::getItemText()</a>	0
<i>get item text</i>	
<a href="#">GridViewControlBase::onPost()</a>	0
<i>handle post events</i>	
<a href="#">GridViewControlBase::onRender()</a>	0
<i>handle request events</i>	
<a href="#">GridViewControlBase::escape()</a>	0
<i>escape</i>	
<a href="#">GridViewControlBase::\$pkey</a>	0
<i>primary key</i>	
<a href="#">GridViewControlBase::\$escapeOutput</a>	0
<i>determines whether to escape the output</i>	
<a href="#">GridViewControlBase::\$parameter</a>	0
<i>event request parameter</i>	
<a href="#">GridViewControlBase::onRequest()</a>	0
<i>handle request events</i>	
<a href="#">GridViewDropDownMenu</a>	0

<i>Represents a GridView DropDownMenu</i>	0
<a href="#"><u>GridViewTextBox</u></a>	0
<i>Represents a GridView TextBox</i>	
<a href="#"><u>GridViewTextBox::\$Size</u></a>	0
<i>size</i>	
<a href="#"><u>GridViewTextBox::getItemText()</u></a>	0
<i>get item text</i>	
<a href="#"><u>GridViewLink</u></a>	0
<i>Represents a GridView link</i>	
<a href="#"><u>GridViewImage</u></a>	0
<i>Represents a GridView image</i>	
<a href="#"><u>GridViewDropDownMenu::\$Items</u></a>	0
<i>items</i>	
<a href="#"><u>GridViewDropDownMenu::getItemText()</u></a>	0
<i>get item text</i>	
<a href="#"><u>GridViewControlBase::\$ajaxPostBack</u></a>	0
<i>event request parameter</i>	
<a href="#"><u>GridViewControlBase</u></a>	0
<i>Represents a GridView control</i>	
<a href="#"><u>GridViewColumnCollection</u></a>	0
<i>Represents a Collection of GridViewColumn objects</i>	
<a href="#"><u>GridViewColumnCollection::add()</u></a>	0
<i>add GridViewColumn to Collection</i>	
<a href="#"><u>GridViewColumnCollection::addBefore()</u></a>	0
<i>add GridViewColumn to Collection before</i>	
<a href="#"><u>GridViewColumnAjaxPostEventHandler</u></a>	0
<i>Provides event handling</i>	
<a href="#"><u>GridViewColumnAjaxPostEvent</u></a>	0
<i>Provides event handling</i>	
<a href="#"><u>GridViewColumn::onRender()</u></a>	0
<i>handle render events</i>	
<a href="#"><u>GridViewColumn::onRequest()</u></a>	0
<i>handle request events</i>	
<a href="#"><u>GridViewColumnCollection::findColumn()</u></a>	0
<i>returns index if value is found in collection</i>	
<a href="#"><u>GridViewColumnCollection::itemAt()</u></a>	0
<i>return GridViewColumn at a specified index</i>	
<a href="#"><u>GridViewColumnPostEvent</u></a>	0
<i>Provides event handling</i>	
<a href="#"><u>GridViewColumnPostEventHandler</u></a>	0
<i>Provides event handling</i>	
<a href="#"><u>GridViewColumnCollection::onRequest()</u></a>	0
<i>handle request events</i>	
<a href="#"><u>GridViewColumnCollection::onRender()</u></a>	0
<i>handle render events</i>	
<a href="#"><u>GridViewColumnCollection::onLoad()</u></a>	0
<i>handle load events</i>	
<a href="#"><u>GridViewColumnCollection::onPost()</u></a>	0
<i>handle post events</i>	
<a href="#"><u>GridView::findColumn()</u></a>	0
<i>returns GridViewColumn if column is found in Collection</i>	
<a href="#"><u>GridView::addColumn()</u></a>	0
<i>add a bound column</i>	

<a href="#">GridViewController::make()</a>	0
<i>make</i>	
<a href="#">gridviewcolumnajaxpostevent.class.php</a>	0
<a href="#">gridviewcolumnajaxposteventhandler.class.php</a>	0
<a href="#">GridViewController::ModelNamespace</a>	0
<i>Default namespace</i>	
<a href="#">GridViewController::DefaultBaseController</a>	0
<i>DefaultBaseController</i>	
<a href="#">GridViewController</a>	0
<i>Provides functionality to generate controller/view files</i>	
<a href="#">GridViewController::ControllerNamespace</a>	0
<i>Default namespace</i>	
<a href="#">gridviewcolumnpostevent.class.php</a>	0
<a href="#">gridviewcolumnposteventhandler.class.php</a>	0
<a href="#">gridviewcolumncollection.class.php</a>	0
<a href="#">gridviewcontrolbase.class.php</a>	0
<a href="#">gridviewdropdownmenu.class.php</a>	0
<a href="#">gridviewcolumn.class.php</a>	0
<a href="#">gridviewcheckbox.class.php</a>	0
<a href="#">gridview.class.php</a>	0
<a href="#">gridviewbutton.class.php</a>	0
<a href="#">gridviewController.php</a>	0
<a href="#">Get::handleException()</a>	0
<i>event triggered by an uncaught Exception thrown in the application, can be overridden to provide error handling.</i>	
<a href="#">gridviewsorteventhandler.class.php</a>	0
<a href="#">GridViewChangeEvent</a>	0
<i>Provides event handling</i>	
<a href="#">GridViewChangeEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">gridviewsortevent.class.php</a>	0
<a href="#">gridviewfiltereventhandler.class.php</a>	0
<a href="#">gridviewchangeordereventhandler.class.php</a>	0
<a href="#">gridviewfilterevent.class.php</a>	0
<a href="#">GridViewFilterEvent</a>	0
<i>Provides event handling</i>	
<a href="#">GridViewFilterEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">Get::execute()</a>	0
<i>execute the application</i>	
<a href="#">Get::getOptions()</a>	0
<i>retrieve command line options</i>	
<a href="#">Get::handleError()</a>	0
<i>event triggered by an error in the application, can be overridden to provide error handling.</i>	
<a href="#">Get</a>	0
<i>deplomyent script</i>	
<a href="#">get.class.php</a>	0
<a href="#">GridViewSortEvent</a>	0
<i>Provides event handling</i>	
<a href="#">GridViewSortEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">gridviewimage.class.php</a>	0
<a href="#">gridviewlink.class.php</a>	0

<a href="#">GridView::\$page</a>	0
current grid page	
<a href="#">GridView::\$pageSize</a>	0
Number of rows to display per page, Default is 20	
<a href="#">GridView::\$selected</a>	0
array of values for selected items (used when showList is true)	
<a href="#">GridView::\$orderByField</a>	0
specifies the order by field	
<a href="#">GridView::\$onmouseover</a>	0
specifies the action to take on mouseover events	
<a href="#">GridView::\$ondblclick</a>	0
specifies the action to take on double click events	
<a href="#">GridView::\$onmouseout</a>	0
specifies the action to take on onmouseout events	
<a href="#">GridView::\$showFilters</a>	0
Set to display column filters, Default is false	
<a href="#">GridView::\$showFooter</a>	0
Set to display table footer, Default is false	
<a href="#">GridView::\$sortBy</a>	0
current grid sorting field	
<a href="#">GridView::\$sortOrder</a>	0
current grid sorting order	
<a href="#">GridView::\$valueField</a>	0
specifies the value field	
<a href="#">GridView::\$showPrimaryKey</a>	0
Set to display primary key in view, Default is false	
<a href="#">GridView::\$showPageNumber</a>	0
Set to display table footer, Default is true	
<a href="#">GridView::\$showHeader</a>	0
Set to display column headers, Default is true	
<a href="#">GridView::\$showList</a>	0
Show list view, default is false	
<a href="#">GridView::\$onclick</a>	0
specifies the action to take on click events	
<a href="#">GridView::\$multiple</a>	0
specifies if list can have multiple values (used when showList is true), default is false	
<a href="#">GridView::\$autoGenerateColumns</a>	0
Specifies whether to generate columns from datasource, default is false	
<a href="#">GridView::\$canChangeOrder</a>	0
Specifies if table order can be changed, Default is false	
<a href="#">GridView::\$canSort</a>	0
Specifies if table is sortable, Default is true	
<a href="#">GridView::\$ajaxPostBack</a>	0
Specifies whether form will submit ajax postback on change, Default is false	
<a href="#">GridView::\$actions</a>	0
Array of actions	
<a href="#">gridviewtextbox.class.php</a>	0
<a href="#">GridView</a>	0
Represents a GridView Control	
<a href="#">GridView::\$caption</a>	0
Table caption, Default is none	
<a href="#">GridView::\$columns</a>	0
collection of columns	

<a href="#">GridView::\$moveCursor</a>	0
current grid moveCursor field	
<a href="#">GridView::\$moveOrder</a>	0
current grid moveCursor order	
<a href="#">GridView::\$listName</a>	0
optional name for the list	
<a href="#">GridView::\$filterValues</a>	0
grid filter values	
<a href="#">GridView::\$filterGroupSize</a>	0
Number of records to display in dropdown filter, Default is 10	
<a href="#">GridView::\$filters</a>	0
current grid filters	
<a href="#">gridviewchangeorderevent.class.php</a>	0

## H

<a href="#">HttpResponse::redirect()</a>	0
redirect to new URL	
<a href="#">HttpResponse::setCookie()</a>	0
set cookie	
<a href="#">HttpResponse::write()</a>	0
sends information to an output stream	
<a href="#">htmltestreporter.class.php</a>	0
<a href="#">HttpResponse::getResponseHeaders()</a>	0
return response headers	
<a href="#">HttpResponse::getResponseContent()</a>	0
return response content	
<a href="#">HttpResponse::addHeader()</a>	0
adds header to output stream	
<a href="#">HttpResponse::clear()</a>	0
clear all content from output buffer	
<a href="#">HttpResponse::end()</a>	0
send response to client and stop execution	
<a href="#">HttpResponse::flush()</a>	0
send all content to client	
<a href="#">HTMLTestReporter</a>	0
Provides HTML reporting for test cases	
<a href="#">HTMLTestReporter::paintException()</a>	0
Paints a PHP exception.	
<a href="#">HyperLink::\$target</a>	0
specifies hyperlink target	
<a href="#">HyperLink::\$text</a>	0
specifies hyperlink text	
<a href="#">HyperLink::\$url</a>	0
specifies hyperlink URL	
<a href="#">HyperLink::getDomObject()</a>	0
getDomObject	
<a href="#">HyperLink::\$src</a>	0
specifies hyperlink image source	
<a href="#">HyperLink</a>	0
Represents a Button Control	
<a href="#">HTMLTestReporter::paintFooter()</a>	0



<i>Paints the end of the test with a summary of the passes and failures.</i>	0
<a href="#">HTMLTestReporter::paintHeader()</a>	0
<i>Paints the top of the web page setting the title to the name of the starting test.</i>	
<a href="#">HTMLTestReporter::setElapsedTime()</a>	0
<i>set elapsed time</i>	
<a href="#">hyperlink.class.php</a>	0
<a href="#">HTTPResponse</a>	0
<i>Provides a way to send HTTP response information to the client</i>	
<a href="#">HTTPRequest::getRequestURI()</a>	0
<i>return the URI</i>	
<a href="#">HTTPWebRequest::sendRequest()</a>	0
<i>sends request and closes Stream object</i>	
<a href="#">HTTPWebRequest::setData()</a>	0
<i>set data from string or array</i>	
<a href="#">HTTPWebResponse</a>	0
<i>Represents the response from a HTTPWebRequest message</i>	
<a href="#">HTTPWebResponse::readFromStream()</a>	0
<i>read data from Stream object</i>	
<a href="#">HTTPWebRequest::getResponse()</a>	0
<i>sends request and creates HTTPWebResponse object before closing Stream object</i>	
<a href="#">HTTPWebRequest::getRequestString()</a>	0
<i>returns a string for request</i>	
<a href="#">httpwebresponse.class.php</a>	0
<a href="#">HTTPWebRequest</a>	0
<i>Represents an HTTP request (supports SSL)</i>	
<a href="#">HTTPWebRequest::addHeader()</a>	0
<i>add header to request</i>	
<a href="#">HTTPWebRequest::getRequestStream()</a>	0
<i>returns a Stream object for request</i>	
<a href="#">httprequest.class.php</a>	0
<a href="#">httpresponse.class.php</a>	0
<a href="#">HTTPRequest::getRawPostData()</a>	0
<i>return raw post data</i>	
<a href="#">HTTPRequest::getRequestHeaders()</a>	0
<i>return request headers</i>	
<a href="#">HTTPRequest::getRequestMethod()</a>	0
<i>return the request method</i>	
<a href="#">HTTPRequest::getRequestReferer()</a>	0
<i>return the request method</i>	
<a href="#">HTTPRequest::getQueryString()</a>	0
<i>return the URL</i>	
<a href="#">HTTPRequest::\$request</a>	0
<i>var to hold request vars</i>	
<a href="#">HTTPRequest</a>	0
<i>Provides a way to access HTTP Request information from the client</i>	
<a href="#">HTTPRequest::\$cookie</a>	0
<i>var to hold cookie vars</i>	
<a href="#">HTTPRequest::\$get</a>	0
<i>var to hold get vars</i>	
<a href="#">HTTPRequest::\$post</a>	0
<i>var to hold post vars</i>	

[httpwebrequest.class.php](#) . . . . . 0

# I

[InputBase::\\$validators](#) . . . . . 0

*contains a collection of validators*

[InputBase::\\$stooltip](#) . . . . . 0

*specifies control tool tip*

[InputBase::\\$tabIndex](#) . . . . . 0

*Specifies the tab order if the control*

[InputBase::\\$value](#) . . . . . 0

*Gets or sets value of control*

[InputBase::addValidator\(\)](#) . . . . . 0

*adds a validator to the control*

[InputBase::fetchError\(\)](#) . . . . . 0

*returns the error message element*

[InputBase::error\(\)](#) . . . . . 0

*renders error message if control does not validate*

[InputBase::\\$submitted](#) . . . . . 0

*Specifies whether the data has been submitted*

[InputBase::\\$readonly](#) . . . . . 0

*Specifies whether the control is readonly, Default is false*

[InputBase::\\$dataField](#) . . . . . 0

*Name of the data field in the datasource*

[InputBase::\\$changed](#) . . . . . 0

*specifies whether the control has changed*

[InputBase::\\$autoPostBack](#) . . . . . 0

*Specifies whether form will submit postback on change, Default is false*

[InputBase::\\$defaultHTMLControlId](#) . . . . . 0

*specifies the id of the default html control*

[InputBase::\\$disabled](#) . . . . . 0

*Specifies whether the control is disabled, Default is false*

[InputBase::\\$label](#) . . . . . 0

*Specifies control label*

[InputBase::\\$enableViewState](#) . . . . . 0

*specifies whether the server control persists its view state, Default is true*

[InputBase::fillDataSource\(\)](#) . . . . . 0

*update data source with Control value*

[InputBase::focus\(\)](#) . . . . . 0

*sets focus to the control*

[InputBase::validate\(\)](#) . . . . . 0

*validates control data, returns true on success*

[InputBase::readDataSource\(\)](#) . . . . . 0

*update Control value with data from the data source*

[InputBase::onUpdateAjax\(\)](#) . . . . . 0

*Event called on ajax callback*

[InputChangeEvent](#) . . . . . 0

*Provides event handling*

[InputChangeEventHandler](#) . . . . . 0

*Provides event handling*

[InputPostBackEventHandler](#) . . . . . 0

*Provides event handling*

<a href="#">InputPostEvent</a>	0
<i>Provides event handling</i>	
<a href="#">InputBase::onSaveViewState()</a>	0
<i>write view state to session</i>	
<a href="#">InputBase::onRequest()</a>	0
<i>process the HTTP request array</i>	
<a href="#">InputBase::onDataBind()</a>	0
<i>bind control to datasource</i>	
<i>gets record from dataobject and sets the control value to datafield value</i>	
<a href="#">InputBase::getInputDomObject()</a>	0
<i>returns an input DomObject representing control</i>	
<a href="#">InputBase::getDomObject()</a>	0
<i>returns an input DomObject representing control</i>	
<a href="#">InputBase::onInit()</a>	0
<i>called when control is initiated</i>	
<a href="#">InputBase::onLoad()</a>	0
<i>called when control is loaded</i>	
<a href="#">InputBase::onPost()</a>	0
<i>handle post events</i>	
<a href="#">InputBase::onLoadViewState()</a>	0
<i>read view state from session</i>	
<a href="#">InputBase::\$ajaxValidation</a>	0
<i>Specifies whether form will submit ajax validation, Default is false</i>	
<a href="#">InputBase::\$ajaxPostBack</a>	0
<i>Specifies whether form will submit ajax postback on change, Default is false</i>	
<a href="#">IOException</a>	0
<i>Represents an I/O Exception</i>	
<a href="#">ioexception.class.php</a>	0
<a href="#">IteratorBase::toArray()</a>	0
<i>convert Iterator object into array</i>	
<a href="#">imailclient.class.php</a>	0
<a href="#">IMailClient</a>	0
<i>Provides the interface for Mail Clients</i>	
<a href="#">integervalidator.class.php</a>	0
<a href="#">IMailClient::send()</a>	0
<i>sends a single email to all addresses in message</i>	
<a href="#">IteratorBase::\$items</a>	0
<i>internal array</i>	
<a href="#">IteratorBase</a>	0
<i>Provides the base functionality for creating Iterators</i>	
<a href="#">invalidoperationexception.class.php</a>	0
<a href="#">invalidargumentexception.class.php</a>	0
<a href="#">indexoutofrangeexception.class.php</a>	0
<a href="#">IndexOutOfRangeException</a>	0
<i>Represents an Index Out Of Range Exception</i>	
<a href="#">InvalidArgumentException</a>	0
<i>Represents an Invalid Argument Exception</i>	
<a href="#">iteratorbase.class.php</a>	0
<a href="#">InvalidOperationException</a>	0
<i>Represents an Invalid Operation Exception</i>	
<a href="#">IntegerValidator</a>	0
<i>Provides basic validation for web controls</i>	
<a href="#">IntegerValidator::onLoad()</a>	0

on load	0
<a href="#">InputAjaxChangeEvent</a>	0
Provides event handling	
<a href="#">inputbase.class.php</a>	0
<a href="#">inputposteventhandler.class.php</a>	0
<a href="#">InputAjaxChangeEventHandler</a>	0
Provides event handling	
<a href="#">InputAjaxPostEvent</a>	0
Provides event handling	
<a href="#">InputBase</a>	0
Provides base functionality for Input Controls	
<a href="#">InputAjaxPostEventHandler</a>	0
Provides event handling	
<a href="#">inputpostevent.class.php</a>	0
<a href="#">inputchangeeventhandler.class.php</a>	0
<a href="#">inputajaxchangeevent.class.php</a>	0
<a href="#">IntegerValidator::validate()</a>	0
sets the controlId and prepares the control attributes	
<a href="#">inputajaxchangeeventhandler.class.php</a>	0
<a href="#">inputajaxpostevent.class.php</a>	0
<a href="#">inputchangeevent.class.php</a>	0
<a href="#">inputajaxposteventhandler.class.php</a>	0
<a href="#">index.php</a>	0
Startup script	
This script creates an instance of the application and executes	

## J

<a href="#">JSONServiceBase::\$options</a>	0
specifies encoding options	
<a href="#">JSONServiceBase::configure()</a>	0
configure the web service	
<a href="#">JSONServiceBase::handle()</a>	0
this method will handle the web service request	
<a href="#">JSONServiceBase</a>	0
This class handles all remote procedure calls for a web service	
<a href="#">jsonservicebase.class.php</a>	0
<a href="#">JSONWebService</a>	0
Provides functionality to generate controller/view files	
<a href="#">JSONWebService::ControllerNamespace</a>	0
Default namespace	
<a href="#">JSONWebService::make()</a>	0
make	
<a href="#">jsonwebservice.php</a>	0

## K

<a href="#">KeyCollection::itemAt()</a>	0
return key at a specified index	
<a href="#">KeyCollection</a>	0
Represents a collection of keys	

[keycollection.class.php](#) . . . . . 0

## L

<a href="#">ListItemCollection</a>	0
<i>Represents a Collection of list items</i>	
<a href="#">ListControlBase::validate()</a>	0
<i>validates control data, returns true on success</i>	
<a href="#">ListItemCollection::add()</a>	0
<i>Adds a new key/item pair to a Dictionary object if key does not already exist</i>	
<a href="#">ListItemCollection::containsItem()</a>	0
<i>Returns a Boolean value that indicates whether a specified value exists in the Dictionary object</i>	
<a href="#">ListItemCollection::indexOfItem()</a>	0
<i>returns index of item found in Dictionary</i>	
<a href="#">ListItemCollection::indexOf()</a>	0
<i>returns index of key found in Dictionary</i>	
<a href="#">ListControlBase::onRequest()</a>	0
<i>process the HTTP request array</i>	
<a href="#">ListControlBase::onLoadViewState()</a>	0
<i>read view state from session</i>	
<a href="#">ListControlBase::\$multiple</a>	0
<i>Specifies whether multiple selections are allowed</i>	
<a href="#">ListControlBase::\$items</a>	0
<i>collection of items</i>	
<a href="#">ListControlBase::\$textField</a>	0
<i>Specifies name of text field in datasource</i>	
<a href="#">ListControlBase::\$valueField</a>	0
<i>Specifies name of value field in datasource</i>	
<a href="#">ListControlBase::onLoad()</a>	0
<i>called when control is loaded</i>	
<a href="#">ListControlBase::onDataBind()</a>	0
<i>bind control to data</i>	
<a href="#">LoginForm</a>	0
<i>Represents a LoginForm</i>	
<a href="#">LoginForm::\$disabledMsg</a>	0
<i>specifies the message when account is disabled</i>	
<a href="#">LoginForm::getDomObject()</a>	0
<i>returns a DomObject representing control</i>	
<a href="#">LoginForm::\$successMsg</a>	0
<i>specifies the message on successful login</i>	
<a href="#">LoginForm::onInit()</a>	0
<i>called when control is initiated</i>	
<a href="#">LoginForm::onRequest()</a>	0
<i>process the HTTP request array</i>	
<a href="#">LoginFormSubmitEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">LoginFormSubmitEvent</a>	0
<i>Provides event handling</i>	
<a href="#">LoginForm::\$passwordResetSentMsg</a>	0
<i>specifies the message when email address does not exist</i>	
<a href="#">LoginForm::\$passwordResetMsg</a>	0
<i>specifies the message when email address does not exist</i>	

<a href="#">LoginForm::\$emailNotFoundMsg</a>	0
<i>specifies the message when email address does not exist</i>	
<a href="#">LoginForm::\$emailFromAddress</a>	0
<i>Specifies the from address when email is sent</i>	
<a href="#">LoginForm::\$emailSubject</a>	0
<i>Specifies the subject line when email is sent</i>	
<a href="#">LoginForm::\$invalidCredentialsMsg</a>	0
<i>specifies the message when login failed</i>	
<a href="#">LoginForm::\$mailClient</a>	0
<i>Specifies the mail client for sending</i>	
<a href="#">LoginForm::\$lockoutMsg</a>	0
<i>specifies the message when account is locked out</i>	
<a href="#">ListControlBase</a>	0
<i>Provides base functionality for List Controls</i>	
<a href="#">ListBox::onUpdateAjax()</a>	0
<i>Event called on ajax callback</i>	
<a href="#">ListController::DefaultBaseController</a>	0
<i>DefaultBaseController</i>	
<a href="#">ListController::ControllerNamespace</a>	0
<i>Default namespace</i>	
<a href="#">ListController::ModelNamespace</a>	0
<i>Default namespace</i>	
<a href="#">ListController::make()</a>	0
<i>make</i>	
<a href="#">LengthValidator</a>	0
<i>Provides basic validation for web controls</i>	
<a href="#">lengthvalidator.class.php</a>	0
<a href="#">ListController</a>	0
<i>Provides functionality to generate controller/view files</i>	
<a href="#">listcontroller.php</a>	0
<a href="#">LoggerBase::flush()</a>	0
<i>This method flushes all messages</i>	
<a href="#">LoggerBase</a>	0
<i>Provides access to read/write log messages</i>	
<a href="#">LoggerBase::log()</a>	0
<i>This method writes a log message to memory</i>	
<a href="#">LoggerBase::logs()</a>	0
<i>This method retrieves all log messages</i>	
<a href="#">LoggerException</a>	0
<i>Represents a Logger Exception</i>	
<a href="#">loggerexception.class.php</a>	0
<a href="#">LengthValidator::onLoad()</a>	0
<i>on load</i>	
<a href="#">LengthValidator::validate()</a>	0
<i>sets the controlId and prepares the control attributes</i>	
<a href="#">Label::\$text</a>	0
<i>specifies button text</i>	
<a href="#">Label</a>	0
<i>Represents a Button Control</i>	
<a href="#">Label::getDomObject()</a>	0
<i>getDomObject</i>	
<a href="#">ListBox</a>	0
<i>Represents a ListBox Control</i>	

<a href="#">ListBox::getDomObject()</a>	0
<i>returns a DomObject representing control</i>	
<a href="#">ListBox::\$listSize</a>	0
<i>Size of listbox, default is 6</i>	
<a href="#">loginform.class.php</a>	0
<a href="#">listitemcollection.class.php</a>	0
<a href="#">loginformsubmiteventhandler.class.php</a>	0
<a href="#">loginformsubmitevent.class.php</a>	0
<a href="#">label.class.php</a>	0
<a href="#">listbox.class.php</a>	0
<a href="#">listcontrolbase.class.php</a>	0
<a href="#">loggerbase.class.php</a>	0

## M

<a href="#">MailMessage::addBcc()</a>	0
<i>add email address to blind copy</i>	
<a href="#">MailMessage</a>	0
<i>Represents an E-Mail Message</i>	
<a href="#">MailMessage::addCc()</a>	0
<i>add email address to copy</i>	
<a href="#">MailMessage::attach()</a>	0
<i>add an attachment to the mail message</i>	
<a href="#">MailMessage::getContent()</a>	0
<i>get message content</i>	
<a href="#">mailmessage.class.php</a>	0
<a href="#">MySQLTransaction::rollbackTransaction()</a>	0
<i>Implements a rollback</i>	
<a href="#">MySQLQueryBuilder::getQuery()</a>	0
<i>get SQL query</i>	
<a href="#">MySQLQueryBuilder::addWhereClause()</a>	0
<i>add where clause</i>	
<a href="#">MySQLTransaction</a>	0
<i>Represents an open connection to a MySQL database</i>	
<a href="#">MySQLTransaction::beginTransaction()</a>	0
<i>Begins a transaction</i>	
<a href="#">MySQLTransaction::commitTransaction()</a>	0
<i>Implements a commit</i>	
<a href="#">MailMessage::getHeaders()</a>	0
<i>get message headers</i>	
<a href="#">MailMessage::setBody()</a>	0
<i>set message body</i>	
<a href="#">model.php</a>	0
<a href="#">makebase.class.php</a>	0
<a href="#">MakeBase</a>	0
<i>Provides the interface for Make scripts</i>	
<a href="#">MakeBase::export()</a>	0
<i>export</i>	
<a href="#">MakeBase::make()</a>	0
<i>make</i>	
<a href="#">MailMessage::setTo()</a>	0
<i>set recipient email address</i>	

<a href="#">MailMessage::setSubject()</a>	0
<i>set message subject</i>	
<a href="#">MailMessage::setCharset()</a>	0
<i>set character set</i>	
<a href="#">MailMessage::setContentType()</a>	0
<i>set content type</i>	
<a href="#">MailMessage::setEncoding()</a>	0
<i>set character encoding</i>	
<a href="#">MailMessage::setFrom()</a>	0
<i>set sender's email address</i>	
<a href="#">MySQLQueryBuilder::addValue()</a>	0
<i>add value</i>	
<a href="#">MySQLQueryBuilder::addTable()</a>	0
<i>add table</i>	
<a href="#">MySQLQueryBuilder::addHavingClause()</a>	0
<i>add having clause</i>	
<a href="#">MySQLQueryBuilder::addGroupByClause()</a>	0
<i>add group by clause</i>	
<a href="#">MySQLQueryBuilder::addJoin()</a>	0
<i>add join</i>	
<a href="#">MySQLQueryBuilder::addOrderByClause()</a>	0
<i>add order by clause</i>	
<a href="#">MySQLQueryBuilder::addTable()</a>	0
<i>add table</i>	
<a href="#">MySQLQueryBuilder::addColumn()</a>	0
<i>add column</i>	
<a href="#">MySQLQueryBuilder</a>	0
<i>Represents an SQL Query</i>	
<a href="#">MySQLDataAdapter::opened()</a>	0
<i>returns true if a connection to a datasource is currently open</i>	
<a href="#">MySQLDataAdapter::open()</a>	0
<i>opens a connection to a mysql database</i>	
<a href="#">MySQLDataAdapter::query()</a>	0
<i>Executes a query procedure on the current connection and return the result</i>	
<a href="#">MySQLDataAdapter::queryBuilder()</a>	0
<i>creates a QueryBuilder object</i>	
<a href="#">MySQLDataAdapter::update()</a>	0
<i>attempt to update a record in the datasource</i>	
<a href="#">MySQLQueryBuilder::addValue()</a>	0
<i>add value</i>	
<a href="#">MySQLQueryBuilder::addWhereClause()</a>	0
<i>add where clause</i>	
<a href="#">MySQLQueryBuilder::addGroupByClause()</a>	0
<i>add group by clause</i>	
<a href="#">MySQLQueryBuilder::addColumn()</a>	0
<i>add column</i>	
<a href="#">MySQLQueryBuilder::addHavingClause()</a>	0
<i>add having clause</i>	
<a href="#">MySQLQueryBuilder::addJoin()</a>	0
<i>add join</i>	
<a href="#">MySQLQueryBuilder::addOrderByClause()</a>	0
<i>add order by clause</i>	
<a href="#">MySQLQueryBuilder</a>	0



<i>Represents an SQL Query</i>	0
<a href="#">MySQLiTransaction::rollbackTransaction()</a>	0
<i>Implements a rollback</i>	
<a href="#">MySQLiQueryBuilder::getQuery()</a>	0
<i>get SQL query</i>	
<a href="#">MySQLiTransaction</a>	0
<i>Represents an open connection to a MySQL database</i>	
<a href="#">MySQLiTransaction::beginTransaction()</a>	0
<i>Begins a transaction</i>	
<a href="#">MySQLiTransaction::commitTransaction()</a>	0
<i>Implements a commit</i>	
<a href="#">Model</a>	0
<i>Provides functionality to generate a model file</i>	
<a href="#">Model::ModelNamespace</a>	0
<i>Default namespace</i>	
<a href="#">masterviewiniteventhandler.class.php</a>	0
<a href="#">masterviewinitevent.class.php</a>	0
<a href="#">masterviewloadevent.class.php</a>	0
<a href="#">masterviewloadeventhandler.class.php</a>	0
<a href="#">masterview.class.php</a>	0
<a href="#">MatchValidator::validate()</a>	0
<i>sets the controlId and prepares the control attributes</i>	
<a href="#">MatchValidator::onLoad()</a>	0
<i>on load</i>	
<a href="#">MigrationsTestCaseBase::xxx()</a>	0
<i>returns the response as XMLEntity object</i>	
<a href="#">MigrationsTestCaseBase::tearDown()</a>	0
<i>clean test module</i>	
<a href="#">matchvalidator.class.php</a>	0
<a href="#">MatchValidator</a>	0
<i>Provides basic validation for web controls</i>	
<a href="#">MatchValidator::\$controlToMatch</a>	0
<i>control to match</i>	
<a href="#">MasterView</a>	0
<i>Represents a MasterView Control</i>	
<a href="#">MasterView::content()</a>	0
<i>render content</i>	
<a href="#">MasterViewInitEvent</a>	0
<i>Provides event handling</i>	
<a href="#">MasterView::saveMasterViewViewState()</a>	0
<i>write view state to session</i>	
<a href="#">MasterViewInitEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">MasterViewLoadEvent</a>	0
<i>Provides event handling</i>	
<a href="#">MasterViewLoadEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">MasterView::masterViewRequestProcessor()</a>	0
<i>process the HTTP request array</i>	
<a href="#">MasterView::loadMasterViewViewState()</a>	0
<i>read view state from session</i>	
<a href="#">MasterView::getContentAreaIdString()</a>	0
<i>get content area id string</i>	

<a href="#">MasterView::handleMasterViewPostEvents()</a>	0
<i>handle post events</i>	
<a href="#">MasterView::initMasterView()</a>	0
<i>called when control is initiated</i>	
<a href="#">MasterView::loadMasterView()</a>	0
<i>called when all controls are loaded</i>	
<a href="#">MigrationsTestCaseBase::setUp()</a>	0
<i>setup test module</i>	
<a href="#">MigrationsTestCaseBase::\$state</a>	0
<i>states</i>	
<a href="#">Migrate::getOptions()</a>	0
<i>retrieve command line options</i>	
<a href="#">Migrate::execute()</a>	0
<i>execute the application</i>	
<a href="#">Migrate::handleError()</a>	0
<i>event triggered by an error in the application, can be overridden to provide error handling.</i>	
<a href="#">Migrate::handleException()</a>	0
<i>event triggered by an uncaught Exception thrown in the application, can be overridden to provide error handling.</i>	
<a href="#">MigrationBase</a>	0
<i>Provides base functionality for database migration</i>	
<a href="#">Migrate</a>	0
<i>deplomyent script</i>	
<a href="#">migrations.class.php</a>	0
<a href="#">Model::make()</a>	0
<i>make</i>	
<a href="#">migrate.class.php</a>	0
<a href="#">migrationbase.class.php</a>	0
<a href="#">migrationcompare.class.php</a>	0
<a href="#">MigrationBase::\$db</a>	0
<i>database</i>	
<a href="#">MigrationBase::\$version</a>	0
<i>version</i>	
<a href="#">Migrations::to()</a>	0
<i>set database version</i>	
<a href="#">Migrations::getCurrentVersion()</a>	0
<i>get database version</i>	
<a href="#">Migrations::upgrade()</a>	0
<i>upgrade database</i>	
<a href="#">migrationtestcasebase.class.php</a>	0
<a href="#">MigrationsTestCaseBase</a>	0
<i>Provides base functionality for the UnitTestCase</i>	
<a href="#">Migrations::downgrade()</a>	0
<i>downgrade database</i>	
<a href="#">Migrations</a>	0
<i>Provides base functionality for database migration</i>	
<a href="#">MigrationBase::down()</a>	0
<i>down migration</i>	
<a href="#">MigrationBase::up()</a>	0
<i>up migration</i>	
<a href="#">MigrationCompare</a>	0
<i>Provides migration sorting</i>	
<a href="#">MigrationCompare::compareVersion()</a>	0

<i>method compares version numbers</i>	0
<a href="#">MySQLiDataAdapter::insert()</a>	0
<i>attempt to insert a record into the datasource</i>	
<a href="#">MySQLiDataAdapter::getLastInsertId()</a>	0
<i>return id of last record inserted</i>	
<a href="#">MSSQLDataAdapter::begin()</a>	0
<i>Executes a sql query or procedure on the current connection</i>	
<a href="#">MSSQLDataAdapter::alterTableSchema()</a>	0
<i>alters a TableSchema object</i>	
<a href="#">MSSQLDataAdapter::beginTransaction()</a>	0
<i>creates a Transaction object</i>	
<a href="#">MSSQLDataAdapter::buildSchema()</a>	0
<i>builds a DataBaseSchema object</i>	
<a href="#">MSSQLDataAdapter::close()</a>	0
<i>closes an open connection</i>	
<a href="#">MSSQLDataAdapter::addTableSchema()</a>	0
<i>creates a TableSchema object</i>	
<a href="#">MSSQLDataAdapter::\$charset</a>	0
<i>Specifies the character set</i>	
<a href="#">mysqlDataAdapter.class.php</a>	0
<a href="#">mysqlTransaction.class.php</a>	0
<a href="#">mysqlQueryBuilder.class.php</a>	0
<a href="#">mysqlTransaction.class.php</a>	0
<a href="#">MSSQLDataAdapter</a>	0
<i>Represents an open connection to a MSSQL database</i>	
<a href="#">MSSQLDataAdapter::commit()</a>	0
<i>commit transaction</i>	
<a href="#">MSSQLDataAdapter::delete()</a>	0
<i>attempt to delete a record in the datasource</i>	
<a href="#">MSSQLDataAdapter::getLastInsertId()</a>	0
<i>return id of last record inserted</i>	
<a href="#">MSSQLDataAdapter::getLastColumnIdOfTable()</a>	0
<i>return id of last record inserted for a column in table</i>	
<a href="#">MSSQLDataAdapter::getQuery()</a>	0
<i>Returns SQL based on the QueryBuilder</i>	
<a href="#">MSSQLDataAdapter::getTables()</a>	0
<i>returns a DataSet of tables</i>	
<a href="#">MSSQLDataAdapter::insert()</a>	0
<i>attempt to insert a record into the datasource</i>	
<a href="#">MSSQLDataAdapter::getField()</a>	0
<a href="#">MSSQLDataAdapter::getAffectedRows()</a>	0
<i>return affected rows</i>	
<a href="#">MSSQLDataAdapter::dropTableSchema()</a>	0
<i>drops a TableSchema object</i>	
<a href="#">MSSQLDataAdapter::escapeString()</a>	0
<i>Returns escaped string</i>	
<a href="#">MSSQLDataAdapter::executeInternal()</a>	0
<i>Executes a sql query or procedure on the current connection</i>	
<a href="#">MSSQLDataAdapter::fill()</a>	0
<i>fetches DataSet from database string using source string</i>	
<a href="#">mysqlQueryBuilder.class.php</a>	0
<a href="#">mysqlDataAdapter.class.php</a>	0
<a href="#">Make::handleError()</a>	0

event triggered by an error in the application, can be overridden to provide error handling.	
<a href="#">Make::getOptions()</a>	0
retrieve command line options	
<a href="#">Make::handleException()</a>	0
event triggered by an uncaught Exception thrown in the application, can be overridden to provide error handling.	
<a href="#">MethodNotImplementedException</a>	0
Represents a Method Not Implemented Exception	
<a href="#">MethodNotImplementedException::\$message</a>	0
<a href="#">Make::execute()</a>	0
execute the application	
<a href="#">Make::\$namespace</a>	0
namespace	
<a href="#">missingargumentexception.class.php</a>	0
<a href="#">modelbase.class.php</a>	0
<a href="#">make.class.php</a>	0
<a href="#">Make</a>	0
deplomyent script	
<a href="#">MissingArgumentException</a>	0
Represents a Missing Arguments Exception	
<a href="#">ModelBase</a>	0
This class represents a data model.	
<a href="#">MEMCache::put()</a>	0
Stores a value identified by a key into cache.	
<a href="#">MEMCache::get()</a>	0
Retrieves a value from cache with a specified key.	
<a href="#">mssqldataadapter.class.php</a>	0
<a href="#">mssqlquerybuilder.class.php</a>	0
<a href="#">mssqltransaction.class.php</a>	0
<a href="#">MEMCache::flush()</a>	0
Deletes all values from cache.	
<a href="#">MEMCache::clear()</a>	0
Deletes a value with the specified key from cache	
<a href="#">ModelBase::getClass()</a>	0
static method to create new ActiveRecordBase of this type	
<a href="#">ModelBase::toArray()</a>	0
converts the model into an array	
<a href="#">memcache.class.php</a>	0
<a href="#">MEMCache</a>	0
Provides access to read/write cache objects from memcache	
<a href="#">MSSQLDataAdapter::open()</a>	0
opens a connection to a mysql database	
<a href="#">MSSQLDataAdapter::opened()</a>	0
returns true if a connection to a datasource is currently open	
<a href="#">MySQLDataAdapter::open()</a>	0
opens a connection to a mysql database	
<a href="#">MySQLDataAdapter::insert()</a>	0
attempt to insert a record into the datasource	
<a href="#">MySQLDataAdapter::opened()</a>	0
returns true if a connection to a datasource is currently open	
<a href="#">MySQLDataAdapter::query()</a>	0
Executes a query procedure on the current connection and return the result	
<a href="#">MySQLDataAdapter::queryBuilder()</a>	0

creates a <i>QueryBuilder</i> object	0
<a href="#">MySQLDataAdapter::getLastInsertId()</a>	0
return id of last record inserted	
<a href="#">MySQLDataAdapter::getAffectedRows()</a>	0
return affected rows	
<a href="#">MySQLDataAdapter::delete()</a>	0
attempt to delete a record in the datasource	
<a href="#">MySQLDataAdapter::close()</a>	0
closes an open connection	
<a href="#">MySQLDataAdapter::dropTableSchema()</a>	0
drops a <i>TableSchema</i> object	
<a href="#">MySQLDataAdapter::escapeString()</a>	0
Returns escaped string	
<a href="#">MySQLDataAdapter::fill()</a>	0
fills a <i>DataSet</i> object with the current <i>DataAdapter</i>	
<a href="#">MySQLDataAdapter::update()</a>	0
attempt to update a record in the datasource	
<a href="#">MySQLiDataAdapter</a>	0
Represents an open connection to a MySQL database using the <i>mysqli</i> driver	
<a href="#">MySQLiDataAdapter::dropTableSchema()</a>	0
drops a <i>TableSchema</i> object	
<a href="#">MySQLiDataAdapter::delete()</a>	0
attempt to delete a record in the datasource	
<a href="#">MySQLiDataAdapter::escapeString()</a>	0
Returns escaped string	
<a href="#">MySQLiDataAdapter::fill()</a>	0
fills a <i>DataSet</i> object with the current <i>DataAdapter</i>	
<a href="#">MySQLiDataAdapter::getAffectedRows()</a>	0
return affected rows	
<a href="#">MySQLiDataAdapter::close()</a>	0
closes an open connection	
<a href="#">MySQLiDataAdapter::buildSchema()</a>	0
builds a <i>DataBaseSchema</i> object	
<a href="#">MySQLiDataAdapter::\$charset</a>	0
Specifies the character set	
<a href="#">MySQLiDataAdapter::addTableSchema()</a>	0
creates a <i>TableSchema</i> object	
<a href="#">MySQLiDataAdapter::alterTableSchema()</a>	0
alters a <i>TableSchema</i> object	
<a href="#">MySQLiDataAdapter::beginTransaction()</a>	0
creates a <i>Transaction</i> object	
<a href="#">MySQLDataAdapter::buildSchema()</a>	0
builds a <i>DataBaseSchema</i> object	
<a href="#">MySQLDataAdapter::beginTransaction()</a>	0
creates a <i>Transaction</i> object	
<a href="#">MSSQLQueryBuilder::addGroupByClause()</a>	0
add group by clause	
<a href="#">MSSQLQueryBuilder::addColumnn()</a>	0
add column	
<a href="#">MSSQLQueryBuilder::addHavingClause()</a>	0
add having clause	
<a href="#">MSSQLQueryBuilder::addJoin()</a>	0
add join	

<a href="#">MSSQLQueryBuilder::addOrderByClause()</a>	0
<i>add order by clause</i>	
<a href="#">MSSQLQueryBuilder</a>	0
<i>Represents an SQL Query</i>	
<a href="#">MSSQLDataAdapter::update()</a>	0
<i>attempt to update a record in the datasource</i>	
<a href="#">MSSQLDataAdapter::query()</a>	0
<i>Executes a query procedure on the current connection and return the result</i>	
<a href="#">MSSQLDataAdapter::queryBuilder()</a>	0
<i>creates a QueryBuilder object</i>	
<a href="#">MSSQLDataAdapter::rollback()</a>	0
<i>rollback transaction</i>	
<a href="#">MSSQLDataAdapter::setForeignKeyChecks()</a>	0
<i>set foreign key checks</i>	
<a href="#">MSSQLQueryBuilder::addTable()</a>	0
<i>add table</i>	
<a href="#">MSSQLQueryBuilder::addValue()</a>	0
<i>add value</i>	
<a href="#">MySQLDataAdapter</a>	0
<i>Represents an open connection to a MySQL database</i>	
<a href="#">MSSQLTransaction::rollbackTransaction()</a>	0
<i>Implements a rollback</i>	
<a href="#">MySQLDataAdapter::\$charset</a>	0
<i>Specifies the character set</i>	
<a href="#">MySQLDataAdapter::addTableSchema()</a>	0
<i>creates a TableSchema object</i>	
<a href="#">MySQLDataAdapter::alterTableSchema()</a>	0
<i>alters a TableSchema object</i>	
<a href="#">MSSQLTransaction::commitTransaction()</a>	0
<i>Implements a commit</i>	
<a href="#">MSSQLTransaction::beginTransaction()</a>	0
<i>Begins a transaction</i>	
<a href="#">MSSQLQueryBuilder::addWhereClause()</a>	0
<i>add where clause</i>	
<a href="#">MSSQLQueryBuilder::getQuery()</a>	0
<i>get SQL query</i>	
<a href="#">MSSQLQueryBuilder::is_date()</a>	0
<a href="#">MSSQLTransaction</a>	0
<i>Represents an open connection to a MSSQL database</i>	
<a href="#">methodnotimplementedexception.class.php</a>	0

## N

<a href="#">NumericValidator::validate()</a>	0
<i>sets the controlId and prepares the control attributes</i>	
<a href="#">NumericValidator::onLoad()</a>	0
<i>on load</i>	
<a href="#">NumericValidator</a>	0
<i>Provides basic validation for web controls</i>	
<a href="#">numericvalidator.class.php</a>	0

# P

<a href="#">page.class.php</a>	0
<a href="#">pagecontrollerbase.class.php</a>	0
<a href="#">pagerequesteventhandler.class.php</a>	0
<a href="#">pagerequestevent.class.php</a>	0
<a href="#">Page</a>	0
<i>Represents an Page Control</i>	
<a href="#">Page::\$contentType</a>	0
<i>specifies content-Type</i>	
<a href="#">Page::\$onload</a>	0
<i>specify javascript to execute on page load</i>	
<a href="#">Page::\$ObjectElements</a>	0
<i>array of object elements</i>	
<a href="#">Page::\$metaElements</a>	0
<i>array of meta elements</i>	
<a href="#">Page::\$linkElements</a>	0
<i>array of link elements</i>	
<a href="#">pageposteventhandler.class.php</a>	0
<a href="#">pagepostevent.class.php</a>	0
<a href="#">PageController::DefaultBaseController</a>	0
<i>DefaultBaseController</i>	
<a href="#">PageController::ControllerNamespace</a>	0
<i>Default namespace</i>	
<a href="#">PageController</a>	0
<i>Provides functionality to generate controller/view files</i>	
<a href="#">pagecontroller.php</a>	0
<a href="#">PageController::make()</a>	0
<i>make</i>	
<a href="#">patternvalidator.class.php</a>	0
<a href="#">PatternValidator::validate()</a>	0
<i>sets the controlId and prepares the control attributes</i>	
<a href="#">PatternValidator::onLoad()</a>	0
<i>on load</i>	
<a href="#">PatternValidator</a>	0
<i>Provides basic validation for web controls</i>	
<a href="#">Page::\$scriptElements</a>	0
<i>array of script elements</i>	
<a href="#">Page::addLink()</a>	0
<i>add link element</i>	
<a href="#">PageControllerBase::\$theme</a>	0
<i>Specifies the theme for this page</i>	
<a href="#">PageControllerBase::\$page</a>	0
<i>Contains an instance of the Page component</i>	
<a href="#">PageControllerBase</a>	0
<i>This class handles all requests for a specific page. In addition provides access to a Page component to manage any WebControl components.</i>	
<a href="#">Page::onRequest()</a>	0
<i>Event called when request is processed</i>	
<a href="#">PageControllerBase::getView()</a>	0
<i>return view component for rendering</i>	
<a href="#">PageControllerBase::onLoad()</a>	0
<i>Event called when object created</i>	



<a href="#">PageRequestEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">PageRequestEvent</a>	0
<i>Provides event handling</i>	
<a href="#">PagePostEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">PagePostEvent</a>	0
<i>Provides event handling</i>	
<a href="#">Page::onPost()</a>	0
<i>handle post events</i>	
<a href="#">Page::onInit()</a>	0
<i>Event called when control is initiated</i>	
<a href="#">Page::fetch()</a>	0
<i>returns an html string</i>	
<a href="#">Page::clearAjaxBuffer()</a>	0
<i>clear ajax output buffer</i>	
<a href="#">Page::addScript()</a>	0
<i>add script element</i>	
<a href="#">Page::addMetaTag()</a>	0
<i>add metatag</i>	
<a href="#">Page::fetchPageHeader()</a>	0
<i>return page header</i>	
<a href="#">Page::getAjaxBuffer()</a>	0
<i>return ajax output buffer as a string</i>	
<a href="#">Page::loadAjaxJScriptBuffer()</a>	0
<i>load content into ajax jscript output buffer</i>	
<a href="#">Page::loadAjaxBuffer()</a>	0
<i>load content into ajax output buffer</i>	
<a href="#">Page::isAjaxBufferReady()</a>	0
<i>specifies whether the buffer contains output</i>	
<a href="#">PHPMailClient::send()</a>	0
<i>sends a single email to all addresses in message</i>	
<a href="#">PHPMailClient</a>	0
<i>Handles email sending via PHP mail() function</i>	
<a href="#">PDODataAdapter::escapeString()</a>	0
<i>Returns escaped string</i>	
<a href="#">PDODataAdapter::dropTableSchema()</a>	0
<i>drops a TableSchema object</i>	
<a href="#">PDODataAdapter::delete()</a>	0
<i>attempt to delete a record in the datasource</i>	
<a href="#">PDODataAdapter::close()</a>	0
<i>closes an open connection</i>	
<a href="#">PDODataAdapter::fill()</a>	0
<i>fills a DataSet object with the current DataAdapter</i>	
<a href="#">PDODataAdapter::getAffectedRows()</a>	0
<i>return affected rows</i>	
<a href="#">PDODataAdapter::open()</a>	0
<i>opens a connection to a mysql database</i>	
<a href="#">PDODataAdapter::insert()</a>	0
<i>attempt to insert a record into the datasource</i>	
<a href="#">PDODataAdapter::getLastInsertId()</a>	0
<i>return id of last record inserted</i>	
<a href="#">PDODataAdapter::getField()</a>	0



<a href="#">PDODataAdapter::buildSchema()</a>	0
<i>builds a MSSQL DataBaseSchema object</i>	
<a href="#">PDODataAdapter::beginTransaction()</a>	0
<i>creates a Transaction object</i>	
<a href="#">pdodataadapter.class.php</a>	0
<a href="#">PageControllerCreatePageEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">PageControllerCreatePageEvent</a>	0
<i>Provides event handling</i>	
<a href="#">pagecontrollercreatepageeventhandler.class.php</a>	0
<a href="#">pdoquerybuilder.class.php</a>	0
<a href="#">pdotransaction.class.php</a>	0
<a href="#">PDODataAdapter::alterTableSchema()</a>	0
<i>alters a TableSchema object</i>	
<a href="#">PDODataAdapter::addTableSchema()</a>	0
<i>creates a TableSchema object</i>	
<a href="#">PDODataAdapter</a>	0
<i>Represents an open connection to a MySQL database using the mysqli driver</i>	
<a href="#">PDODataAdapter::opened()</a>	0
<i>returns true if a connection to a datasource is currently open</i>	
<a href="#">PDODataAdapter::query()</a>	0
<i>Executes a query procedure on the current connection and return the result</i>	
<a href="#">PDOQueryBuilder::is_date()</a>	0
<a href="#">PDOQueryBuilder::getQuery()</a>	0
<i>get SQL query</i>	
<a href="#">PDOQueryBuilder::getMySQLQuery()</a>	0
<i>get MySQL SQL query</i>	
<a href="#">PDOQueryBuilder::getMSSQLQuery()</a>	0
<i>get MSSQL SQL query</i>	
<a href="#">PDOTransaction</a>	0
<i>Represents an open connection to a PDO</i>	
<a href="#">PDOTransaction::beginTransaction()</a>	0
<i>Begins a transaction</i>	
<a href="#">phpmailclient.class.php</a>	0
<a href="#">PDOTransaction::rollbackTransaction()</a>	0
<i>Implements a rollback</i>	
<a href="#">PDOTransaction::commitTransaction()</a>	0
<i>Implements a commit</i>	
<a href="#">PDOQueryBuilder::addWhereClause()</a>	0
<i>add where clause</i>	
<a href="#">PDOQueryBuilder::addValue()</a>	0
<i>add value</i>	
<a href="#">PDOQueryBuilder::addColumn()</a>	0
<i>add column</i>	
<a href="#">PDOQueryBuilder</a>	0
<i>Represents an SQL Query</i>	
<a href="#">PDODataAdapter::update()</a>	0
<i>attempt to update a record in the datasource</i>	
<a href="#">PDODataAdapter::queryBuilder()</a>	0
<i>creates a QueryBuilder object</i>	
<a href="#">PDOQueryBuilder::addGroupByClause()</a>	0
<i>add group by clause</i>	
<a href="#">PDOQueryBuilder::addHavingClause()</a>	0

<i>add having clause</i>	
<a href="#">PDOQueryBuilder::addTable()</a>	0
<i>add table</i>	
<a href="#">PDOQueryBuilder::addOrderByClause()</a>	0
<i>add order by clause</i>	
<a href="#">PDOQueryBuilder::addJoin()</a>	0
<i>add join</i>	
<a href="#">pagecontrollercreatepageevent.class.php</a>	0

## Q

<a href="#">QueryBuilderBase::openDataSet()</a>	0
<i>open a DataSet</i>	
<a href="#">QueryBuilderBase::orderBy()</a>	0
<i>impliments `order by` statement</i>	
<a href="#">QueryBuilderBase::rightJoin()</a>	0
<i>impliments `right join` statement</i>	
<a href="#">QueryBuilderBase::leftJoin()</a>	0
<i>impliments `left join` statement</i>	
<a href="#">QueryBuilderBase::insertInto()</a>	0
<i>impliments `insert into` statement</i>	
<a href="#">QueryBuilderBase::having()</a>	0
<i>impliments `having` statement</i>	
<a href="#">QueryBuilderBase::innerJoin()</a>	0
<i>impliments `inner join` statement</i>	
<a href="#">QueryBuilderBase::runQuery()</a>	0
<i>run query</i>	
<a href="#">QueryBuilderBase::select()</a>	0
<i>impliments `select` statement</i>	
<a href="#">QueryBuilderBase::values()</a>	0
<i>impliments `values` statement</i>	
<a href="#">QueryBuilderBase::where()</a>	0
<i>impliments `where` statement</i>	
<a href="#">QueryException</a>	0
<i>Represents a QueryBuilder Exception</i>	
<a href="#">QueryBuilderBase::update()</a>	0
<i>impliments `update` statement</i>	
<a href="#">QueryBuilderBase::truncate()</a>	0
<i>impliments `truncate` statement</i>	
<a href="#">QueryBuilderBase::set()</a>	0
<i>impliments `set` statement</i>	
<a href="#">QueryBuilderBase::setColumns()</a>	0
<i>impliments `set` statement</i>	
<a href="#">QueryBuilderBase::groupBy()</a>	0
<i>impliments `group by` statement</i>	
<a href="#">QueryBuilderBase::getQuery()</a>	0
<i>get query</i>	
<a href="#">QueryBuilderBase::\$statement</a>	0
<i>statement</i>	
<a href="#">QueryBuilderBase::addColumn()</a>	0
<i>add column</i>	
<a href="#">QueryBuilderBase::addGroupByClause()</a>	0

<i>add group by clause</i>	0
<a href="#">QueryBuilderBase::\$empty</a>	0
<i>specifies whether to return empty resultset</i>	
<a href="#">QueryBuilderBase::\$dataAdapter</a>	0
<i>Contains a reference to a DataAdapter object</i>	
<a href="#">queryexception.class.php</a>	0
<a href="#">QueryBuilderBase</a>	0
<i>Represents an database query</i>	
<a href="#">QueryBuilderBase::addHavingClause()</a>	0
<i>add having clause</i>	
<a href="#">QueryBuilderBase::addJoin()</a>	0
<i>add join</i>	
<a href="#">QueryBuilderBase::delete()</a>	0
<i>impliments `delete` statement</i>	
<a href="#">QueryBuilderBase::from()</a>	0
<i>impliments `from` statement</i>	
<a href="#">QueryBuilderBase::addWhereClause()</a>	0
<i>add where clause</i>	
<a href="#">QueryBuilderBase::addValue()</a>	0
<i>add value</i>	
<a href="#">QueryBuilderBase::addOrderByClause()</a>	0
<i>add order by clause</i>	
<a href="#">QueryBuilderBase::addTable()</a>	0
<i>add table</i>	
<a href="#">querybuilderbase.class.php</a>	0

## R

<a href="#">RadioButtonList::\$ondblclick</a>	0
<i>Specifies the action to take on double click events</i>	
<a href="#">RadioButtonList::getDomObject()</a>	0
<i>returns a DomObject representing control</i>	
<a href="#">RadioButtonList::\$onclick</a>	0
<i>Specifies the action to take on click events</i>	
<a href="#">RadioButtonList::\$multiple</a>	0
<i>Size of listbox</i>	
<a href="#">RadioButtonList</a>	0
<i>Represents a RadioButtonList Control</i>	
<a href="#">RadioButtonList::onLoad()</a>	0
<i>called when control is loaded</i>	
<a href="#">RadioGroup</a>	0
<i>Represents a RadioGroup Control</i>	
<a href="#">RadioGroup::onRequest()</a>	0
<i>process the HTTP request array</i>	
<a href="#">ReportDetailsEvent</a>	0
<i>Provides event handling</i>	
<a href="#">RadioGroup::onLoad()</a>	0
<i>called when control is loaded</i>	
<a href="#">RadioGroup::getDomObject()</a>	0
<i>returns a DomObject representing control</i>	
<a href="#">RadioGroup::add()</a>	0
<i>adds child control to collection</i>	

<a href="#">RadioButton::onRequest()</a>	0
<i>process the HTTP request array</i>	
<a href="#">RadioButton::onLoad()</a>	0
<i>called when control is loaded</i>	
<a href="#">radiobutton.class.php</a>	0
<a href="#">radiobuttonlist.class.php</a>	0
<a href="#">restwebservicebase.class.php</a>	0
<a href="#">reportheadereventhandler.class.php</a>	0
<a href="#">reportheaderevent.class.php</a>	0
<a href="#">radiogroup.class.php</a>	0
<a href="#">reportgroup.class.php</a>	0
<a href="#">RadioButton::getDomObject()</a>	0
<i>returns a DomObject representing control</i>	
<a href="#">RadioButton::\$checked</a>	0
<i>Specifies checked status</i>	
<a href="#">RadioButton</a>	0
<i>Represents a RadioButton Control</i>	
<a href="#">reportview.class.php</a>	0
<a href="#">ReportDetailsEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">ReportFooterEvent</a>	0
<i>Provides event handling</i>	
<a href="#">RESTWebServiceBase</a>	0
<i>This class handles all remote procedure calls for a REST web service</i>	
<a href="#">RESTWebServiceBase::\$options</a>	0
<i>specifies encoding options</i>	
<a href="#">ReportView::onDataBind()</a>	0
<i>bind control to data</i>	
<a href="#">ReportView::getGrouping()</a>	0
<i>return grouping array</i>	
<a href="#">ReportView::getDomObject()</a>	0
<i>returns a DomObject for rendering</i>	
<a href="#">RESTWebServiceBase::configure()</a>	0
<i>configure the server</i>	
<a href="#">RESTWebServiceBase::delete()</a>	0
<i>handle delete requests</i>	
<a href="#">RESTWebServiceBase::put()</a>	0
<i>handle put requests</i>	
<a href="#">RESTWebServiceBase::post()</a>	0
<i>handle post requests</i>	
<a href="#">RESTWebServiceBase::handle()</a>	0
<i>this method will handle the web service request</i>	
<a href="#">RESTWebServiceBase::get()</a>	0
<i>handle get requests</i>	
<a href="#">ReportView::fetch()</a>	0
<i>returns an html string</i>	
<a href="#">ReportView::addSorting()</a>	0
<i>adds a sorting field to the report</i>	
<a href="#">ReportGroupFooterEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">ReportGroupHeaderEvent</a>	0
<i>Provides event handling</i>	
<a href="#">ReportGroupFooterEvent</a>	0

<i>Provides event handling</i>	0
<a href="#">ReportGroup</a>	0
<i>Represents a ReportGroup</i>	
<a href="#">ReportFooterEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">ReportGroupHeaderEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">ReportHeaderEvent</a>	0
<i>Provides event handling</i>	
<a href="#">ReportView::addGrouping()</a>	0
<i>adds a grouping field to the report</i>	
<a href="#">ReportView::addFilter()</a>	0
<i>adds a filter to the report</i>	
<a href="#">ReportView</a>	0
<i>Represents a ReportView Control</i>	
<a href="#">ReportHeaderEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">reportgroupheadereventhandler.class.php</a>	0
<a href="#">reportgroupheaderevent.class.php</a>	0
<a href="#">Rum::sendHTTPError()</a>	0
<i>This method sends an HTTP status message to the client</i>	
<a href="#">Rum::tl()</a>	0
<i>This method translates a message into the appropriate language</i>	
<a href="#">Rum::requestHandler()</a>	0
<i>This method returns the current controller</i>	
<a href="#">Rum::messages()</a>	0
<i>returns the application messages as an HTML element</i>	
<a href="#">Rum::log()</a>	0
<i>This method automatically writes a string to a log stamped with the current time</i>	
<a href="#">Rum::trace()</a>	0
<i>This method traces a variable</i>	
<a href="#">Rum::uri()</a>	0
<i>This method returns a URI based on the requested page and parameters</i>	
<a href="#">RelationshipType::BelongsTo()</a>	0
<i>Specifies the relationship is a many to one</i>	
<a href="#">RelationshipType</a>	0
<i>Represents a relationship type</i>	
<a href="#">relationshiptype.class.php</a>	0
<a href="#">Rum::url()</a>	0
<i>This method returns a URL based on the requested page and parameters</i>	
<a href="#">Rum::link()</a>	0
<i>returns an HTML anchor element</i>	
<a href="#">Rum::forward()</a>	0
<i>This method sets the next URI once the current controller is finished executing, the servlet will re-request the page with the requested action. This method allows you to replace post data with keep friendly urls that can be bookmarked.</i>	
<a href="#">Rum::baseURI()</a>	0
<i>returns the application base uri</i>	
<a href="#">Rum::breadcrumb()</a>	0
<i>returns the current page breadcrumb trail as an HTML element</i>	
<a href="#">Rum::app()</a>	0
<i>This method returns the application</i>	
<a href="#">Rum</a>	0

<i>Provides access to application services</i>	0
<a href="#">rum.class.php</a>	0
<a href="#">Rum::cache()</a>	0
<i>This method returns the configuration object</i>	
<a href="#">Rum::clearMessages()</a>	0
<i>this method removes any displayed messages</i>	
<a href="#">Rum::flash()</a>	0
<i>this method adds message to message stack</i>	
<a href="#">Rum::escape()</a>	0
<i>This method returns an escaped string using the application charset for outputting</i>	
<a href="#">Rum::db()</a>	0
<i>This method returns the application DataAdapter object</i>	
<a href="#">Rum::config()</a>	0
<i>This method returns the configuration object</i>	
<a href="#">RelationshipType::HasMany()</a>	0
<i>Specifies the relationship is a one to many</i>	
<a href="#">RelationshipType::HasManyAndBelongsTo()</a>	0
<i>Specifies the relationship is a many to many</i>	
<a href="#">RequiredValidator::onLoad()</a>	0
<i>on load</i>	
<a href="#">RequiredValidator::validate()</a>	0
<i>validates the control</i>	
<a href="#">RequiredValidator</a>	0
<i>Provides basic validation for web controls</i>	
<a href="#">RangeValidator::validate()</a>	0
<i>sets the controlId and prepares the control attributes</i>	
<a href="#">RangeValidator::onLoad()</a>	0
<i>on load</i>	
<a href="#">reportdetailsevent.class.php</a>	0
<a href="#">reportdetailseventhandler.class.php</a>	0
<a href="#">reportgroupfootereventhandler.class.php</a>	0
<a href="#">reportgroupfooterevent.class.php</a>	0
<a href="#">reportfootereventhandler.class.php</a>	0
<a href="#">reportfooterevent.class.php</a>	0
<a href="#">RangeValidator</a>	0
<i>Provides basic validation for web controls</i>	
<a href="#">requiredvalidator.class.php</a>	0
<a href="#">RESTWebService::ControllerNamespace</a>	0
<i>Default namespace</i>	
<a href="#">RESTWebService::make()</a>	0
<i>make</i>	
<a href="#">RESTWebService</a>	0
<i>Provides functionality to generate controller/view files</i>	
<a href="#">restwebservice.php</a>	0
<a href="#">RelationshipType:: toString()</a>	0
<i>Returns a string representing the type</i>	
<a href="#">roles.class.php</a>	0
<a href="#">Roles</a>	0
<i>Provides application wide authentication</i>	
<a href="#">rangevalidator.class.php</a>	0
<a href="#">Roles::isUserInRoles()</a>	0
<i>returns true if current user is a member of the provided roles</i>	
<a href="#">Roles::isUserInRole()</a>	0

<i>returns true if current user is a member of the provide role</i>	0
<a href="#">Roles::getRolesForUser()</a>	0
<i>get roles for current user</i>	
<a href="#">rum.php</a>	0
<i>Startup script</i>	

## S

<a href="#">StreamBase::setEOS()</a>	0
<i>Sets the current position to be the end of the stream (EOS)</i>	
<a href="#">StreamBase::setBOS()</a>	0
<i>Sets the current position to be the end of the stream (EOS)</i>	
<a href="#">StreamBase::readln()</a>	0
<i>Reads one line from a TextStream file and returns the result</i>	
<a href="#">StreamBase::setPosition()</a>	0
<i>Sets the current position from the beginning of a Stream object</i>	
<a href="#">StreamBase::write()</a>	0
<i>Writes binary data to a binary Stream object</i>	
<a href="#">smtpclient.class.php</a>	0
<a href="#">sendmailclient.class.php</a>	0
<a href="#">StreamBase::writeln()</a>	0
<i>Writes a specified number of new-line character to a TextStream file</i>	
<a href="#">StreamBase::read()</a>	0
<i>Reads the entire stream or a specified number of bytes from a binary Stream object</i>	
<a href="#">StreamBase::length()</a>	0
<i>Returns the size of the data</i>	
<a href="#">StreamBase::\$handle</a>	0
<i>handle to file</i>	
<a href="#">StreamBase</a>	0
<i>Provides base functionality for streaming</i>	
<a href="#">SocketStream::open()</a>	0
<i>opens a socket for input output</i>	
<a href="#">StreamBase::close()</a>	0
<i>closes an open connection</i>	
<a href="#">StreamBase::eos()</a>	0
<i>Returns whether the current position is at the end of the stream or not</i>	
<a href="#">StreamBase::getPosition()</a>	0
<i>Returns the current position from the beginning of a Stream object</i>	
<a href="#">StreamBase::flush()</a>	0
<i>Sends the contents of the Stream buffer to the associated underlying object</i>	
<a href="#">SendMailClient</a>	0
<i>Handles email sending via the \$sendmail program</i>	
<a href="#">SendMailClient::send()</a>	0
<i>sends a single email to all addresses in message</i>	
<a href="#">SOAPWebServiceBase::\$encoding</a>	0
<i>specifies the character set for the soap message</i>	
<a href="#">SOAPWebServiceBase::\$cache</a>	0
<i>specifies whether to cache the WSDL</i>	
<a href="#">SOAPWebServiceBase</a>	0
<i>This class handles all remote procedure calls for a SOAP web service</i>	
<a href="#">SOAPWebServiceBase::\$namespace</a>	0
<i>specifies the namespace</i>	



<a href="#">SOAPWebServiceBase::\$version</a>	0
<i>specifies the SOAP version</i>	
<a href="#">SOAPWebServiceBase::handle()</a>	0
<i>this method will handle the web service request</i>	
<a href="#">SOAPWebServiceBase::configure()</a>	0
<i>configure the web service</i>	
<a href="#">SOAPWebServiceBase::\$reserved_methods</a>	0
<i>array of reserved methods</i>	
<a href="#">soapwebservicebase.class.php</a>	0
<a href="#">SOAPWebService::make()</a>	0
<i>make</i>	
<a href="#">SMTPClient::open()</a>	0
<i>Initiates a connection to an SMTP server operation failed.</i>	
<a href="#">SMTPClient::close()</a>	0
<i>Closes the active SMTP session if one exists.</i>	
<a href="#">SMTPClient</a>	0
<i>Handles email sending via smtp client</i>	
<a href="#">SMTPClient::send()</a>	0
<i>sends a single email to all addresses in message</i>	
<a href="#">soapwebservice.php</a>	0
<a href="#">SOAPWebService::ControllerNamespace</a>	0
<i>Default namespace</i>	
<a href="#">SOAPWebService</a>	0
<i>Provides functionality to generate controller/view files</i>	
<a href="#">SocketStream</a>	0
<i>Provides a way to stream data to and from a socket</i>	
<a href="#">Session::write()</a>	0
<i>write and close the session</i>	
<a href="#">StringDictionary::keyAt()</a>	0
<i>returns the key of a specified index</i>	
<a href="#">StringDictionary::itemAt()</a>	0
<i>returns the value of a specified index</i>	
<a href="#">StringDictionary::item()</a>	0
<i>return the value of a key in the Dictionary</i>	
<a href="#">StringDictionary::trim()</a>	0
<i>trim whitespace for all values</i>	
<a href="#">schemaexception.class.php</a>	0
<a href="#">SSHClientBase</a>	0
<i>Provides base functionality for connecting via SSH</i>	
<a href="#">sshclientbase.class.php</a>	0
<a href="#">SchemaException</a>	0
<i>Represents a Schema Exception</i>	
<a href="#">StringDictionary::add()</a>	0
<i>Adds a new key/item pair to a StringCollection if key does not already exist</i>	
<a href="#">StringDictionary</a>	0
<i>Represents a dictionary of key/value strings</i>	
<a href="#">stringdictionary.class.php</a>	0
<a href="#">stringcollection.class.php</a>	0
<a href="#">strt()</a>	0
<i>Begin timer</i>	
<a href="#">StringCollection</a>	0
<i>Represents a collection of strings</i>	



<a href="#">StringCollection::add()</a>	0
<i>add item to Collection</i>	
<a href="#">StringCollection::trim()</a>	0
<i>trim whitespace for all values</i>	
<a href="#">StringCollection::itemAt()</a>	0
<i>return item at a specified index</i>	
<a href="#">SSHClientBase::\$password</a>	0
<i>Specifies the login password</i>	
<a href="#">SSHClientBase::\$port</a>	0
<i>Specifies the server port</i>	
<a href="#">Session::getSessionData()</a>	0
<i>return session array</i>	
<a href="#">Session::destroy()</a>	0
<i>destroy the session</i>	
<a href="#">Session::close()</a>	0
<i>close the session</i>	
<a href="#">Session::read()</a>	0
<i>initialize and read session data once started (implicit)</i>	
<a href="#">Session::regenerateId()</a>	0
<i>regenerate the session id</i>	
<a href="#">Session::start()</a>	0
<i>This method will start a new session or continue an existing session.</i>	
<a href="#">Session::removeAll()</a>	0
<i>remove all session keys</i>	
<a href="#">Session</a>	0
<i>Provides a way to access the persistant session</i>	
<a href="#">session.class.php</a>	0
<a href="#">SSHClientBase::exec()</a>	0
<i>execute deployment script</i>	
<a href="#">SSHClientBase::\$username</a>	0
<i>Specifies the login username</i>	
<a href="#">SSHClientBase::\$server</a>	0
<i>Specifies the server name</i>	
<a href="#">SSHClientBase::put()</a>	0
<i>put file on remote server</i>	
<a href="#">SSHClientBase::run()</a>	0
<i>execute on remote server</i>	
<a href="#">streambase.class.php</a>	0
<a href="#">socketstream.class.php</a>	0
<a href="#">stp()</a>	0
<i>Stop timer and print results</i>	

## T

<a href="#">TimeSelector::onLoad()</a>	0
<i>called when control is loaded</i>	
<a href="#">TimeSelector::onRequest()</a>	0
<i>process the HTTP request array</i>	
<a href="#">TreeNode</a>	0
<i>Represents a tree node</i>	
<a href="#">TreeNode::\$childNodes</a>	0
<i>collection of child nodes</i>	

<a href="#">TimeSelector::getDomObject()</a>	0
<i>returns widget object</i>	
<a href="#">TimeSelector::\$allowNull</a>	0
<i>specifies whether to allow nulls</i>	
<a href="#">TextBox::onLoad()</a>	0
<i>called when control is loaded</i>	
<a href="#">TextBox::onRequest()</a>	0
<i>process the HTTP request array</i>	
<a href="#">TimeSelector</a>	0
<i>handles date control element creation (string)</i>	
<i>abstracts away the presentation logic and data access layer</i>	
<i>the server-side control for WebWidgets</i>	
<a href="#">TreeNode::\$expanded</a>	0
<i>Specifies if the node is expanded by default</i>	
<a href="#">TreeNode::\$id</a>	0
<i>node id</i>	
<a href="#">TreeNode::collapseAll()</a>	0
<i>collapses all children</i>	
<a href="#">TreeNode::createChild()</a>	0
<i>creates a child TreeNode in the current TreeNode</i>	
<a href="#">TreeNode::expand()</a>	0
<i>expand node</i>	
<a href="#">TreeNode::expandAll()</a>	0
<i>expands all children</i>	
<a href="#">TreeNode::collapse()</a>	0
<i>collapse node</i>	
<a href="#">TreeNode::addChild()</a>	0
<i>adds child TreeNode to current TreeNode</i>	
<a href="#">TreeNode::\$imgSrc</a>	0
<i>specifies the path of the icon image</i>	
<a href="#">TreeNode::\$parentNode</a>	0
<i>contains parent node</i>	
<a href="#">TreeNode::\$textOrHtml</a>	0
<i>node text</i>	
<a href="#">TextBox::getDomObject()</a>	0
<i>returns a DomObject representing control</i>	
<a href="#">TextBox::fillDataSet()</a>	0
<i>update DataSet with Control data</i>	
<a href="#">treenodecollection.class.php</a>	0
<a href="#">treeview.class.php</a>	0
<a href="#">TextArea</a>	0
<i>Represents a TextBox Control</i>	
<a href="#">TextArea::\$multiline</a>	0
<i>Specifies whether textbox will accept multiple lines of text, default is true</i>	
<a href="#">treenode.class.php</a>	0
<a href="#">timeselector.class.php</a>	0
<a href="#">Timer::stop()</a>	0
<i>stop timer</i>	
<a href="#">textarea.class.php</a>	0
<a href="#">textbox.class.php</a>	0
<a href="#">TextBox</a>	0
<i>Represents a TextBox Control</i>	
<a href="#">TextBox::\$cols</a>	0

	<i>Specifies that number of columns in multiline textbox, default is 60</i>	0
<a href="#">TextBox::\$rows</a>		0
	<i>Specifies that number of rows in multiline textbox, default is 5</i>	0
<a href="#">TextBox::\$size</a>		0
	<i>Specifies the size of a textbox, default is 30</i>	0
<a href="#">TextBox::\$watermark</a>		0
	<i>Specifies an optional watermark that is displayed when the control is empty</i>	0
<a href="#">TextBox::\$multiline</a>		0
	<i>Specifies whether textbox will accept multiple lines of text, default is false</i>	0
<a href="#">TextBox::\$maxLength</a>		0
	<i>Max Length of value when set to non zero, default is 0</i>	0
<a href="#">TextBox::\$disableAutoComplete</a>		0
	<i>Specifies whether to disable the browsers auto complete feature, default is false</i>	0
<a href="#">TextBox::\$disableEnterKey</a>		0
	<i>Specifies whether to disable the enter key, default is false</i>	0
<a href="#">TextBox::\$mask</a>		0
	<i>Specifies whether characters will be masked, default is false</i>	0
<a href="#">TreeNode::findChildNode()</a>		0
	<i>returns TreeNode if node is found in Collection</i>	0
<a href="#">TreeNode::getAncestry()</a>		0
	<i>returns node ancestry</i>	0
<a href="#">textnode.class.php</a>		0
<a href="#">TextNode</a>		0
	<i>Represents a TextNode XML entity</i>	0
<a href="#">TextNode::addChild()</a>		0
	<i>add a reference to a child element</i>	0
<a href="#">TextNode::appendAttribute()</a>		0
	<i>append attribute of xml node overwrite attribute with same name</i>	0
<a href="#">TreeView::onSaveViewState()</a>		0
	<i>write view state to session</i>	0
<a href="#">TreeView::onRequest()</a>		0
	<i>process the HTTP request array</i>	0
<a href="#">TreeView::getTreeNodeDomObject()</a>		0
	<i>returns DomObject for the TreeNode and all child TreeNodes</i>	0
<a href="#">TreeView::onLoad()</a>		0
	<i>called when control is loaded</i>	0
<a href="#">TreeView::onLoadViewState()</a>		0
	<i>read view state from session</i>	0
<a href="#">TextNode::getAttribute()</a>		0
	<i>get attribute of xml node</i>	0
<a href="#">TextNode::getChild()</a>		0
	<i>returns a reference to a child element by index</i>	0
<a href="#">TextNode::removeChild()</a>		0
	<i>remove a reference to a child element</i>	0
<a href="#">TextNode::replaceChild()</a>		0
	<i>replace a reference to a child element</i>	0
<a href="#">TextNode::setAttribute()</a>		0
	<i>set attribute</i>	0
<a href="#">TextNode::removeAttribute()</a>		0
	<i>remove an attribute from an xml node</i>	0
<a href="#">TextNode::getXMLString()</a>		0
	<i>return XML string</i>	0

<a href="#">TextNode::getChildByAttribute()</a>	0
returns a reference to a child element by element attribute	
<a href="#">TextNode::getChildByName()</a>	0
returns a reference to a child element by element name	
<a href="#">TextNode::getChildrenByName()</a>	0
returns all children by name	
<a href="#">TreeView::getDomObject()</a>	0
returns a DomObject representing control	
<a href="#">TreeView::findTreeItem()</a>	0
traverses all tree nodes and returns true if a node is found	
<a href="#">TreeNode::hasChildNode()</a>	0
returns true if node is found in Collection	
<a href="#">TreeNode::loadViewState()</a>	0
read view state from session	
<a href="#">TreeNode::requestProcessor()</a>	0
process the HTTP request array	
<a href="#">TreeNode::saveViewState()</a>	0
write view state to session	
<a href="#">TreeNode::hasChild()</a>	0
returns true if child node exists	
<a href="#">TreeNode::getSiblings()</a>	0
returns array of sibling nodes	
<a href="#">TreeNode::getChild()</a>	0
returns child node by Id	
<a href="#">TreeNode::getChildren()</a>	0
returns array of sibling nodes	
<a href="#">TreeNode::getParent()</a>	0
get parent node	
<a href="#">TreeNode::setParent()</a>	0
set parent node	
<a href="#">TreeNodeCollection</a>	0
Represents a Collection of TreeNode objects	
<a href="#">TreeView::\$rootNode</a>	0
contains the root node	
<a href="#">TreeView::\$showIcons</a>	0
specifies whether to show icons in tree, Default is true	
<a href="#">TreeView::\$showRoot</a>	0
specifies whether to show the root node	
<a href="#">TreeView</a>	0
Represents a TreeView Control	
<a href="#">TreeNodeCollection::itemAt()</a>	0
return TreeNode at a specified index	
<a href="#">TreeNodeCollection::add()</a>	0
add TreeNode to Collection	
<a href="#">TreeNodeCollection::contains()</a>	0
returns true if array item is found	
<a href="#">TreeNodeCollection::indexOf()</a>	0
returns index if value is found in collection	
<a href="#">Timer::start()</a>	0
start timer	
<a href="#">Timer::resume()</a>	0
resume the timer	
<a href="#">TextDataAdapter::delete()</a>	0

<i>attempt to delete a record in the datasource</i>	0
<a href="#">TextDataAdapter::dropTableSchema()</a>	0
<i>drops a TableSchema object</i>	
<a href="#">TextDataAdapter::escapeString()</a>	0
<i>Returns escaped string</i>	
<a href="#">TextDataAdapter::fill()</a>	0
<i>fetches DataSet from datasource string using source string</i>	
<a href="#">TextDataAdapter::close()</a>	0
<i>closes an open connection</i>	
<a href="#">TextDataAdapter::buildSchema()</a>	0
<i>builds a DataBaseSchema object</i>	
<a href="#">TextDataAdapter::addTableSchema()</a>	0
<i>creates a TableSchema object</i>	
<a href="#">TextDataAdapter::alterTableSchema()</a>	0
<i>alters a TableSchema object</i>	
<a href="#">TextDataAdapter::beginTransaction()</a>	0
<i>creates a Transaction object</i>	
<a href="#">TextDataAdapter::getAffectedRows()</a>	0
<i>return affected rows</i>	
<a href="#">TextDataAdapter::getLastInsertId()</a>	0
<i>return id of last record inserted</i>	
<a href="#">TextDataAdapter::update()</a>	0
<i>attempt to update a record in the datasource</i>	
<a href="#">TransactionBase</a>	0
<i>Represents a transaction</i>	
<a href="#">TransactionBase::\$dataAdapter</a>	0
<i>Contains a reference to a DataAdapter object</i>	
<a href="#">TransactionBase::begin()</a>	0
<i>Begins a transaction</i>	
<a href="#">TextDataAdapter::queryBuilder()</a>	0
<i>creates a QueryBuilder object</i>	
<a href="#">TextDataAdapter::query()</a>	0
<i>Executes a query procedure on the current connection and return the result</i>	
<a href="#">TextDataAdapter::insert()</a>	0
<i>attempt to insert a record into the datasource</i>	
<a href="#">TextDataAdapter::open()</a>	0
<i>opens a connection to a text file</i>	
<a href="#">TextDataAdapter::opened()</a>	0
<i>returns true if a connection to a datasource is currently open</i>	
<a href="#">TextDataAdapter</a>	0
<i>Represents an open connection to a tab or comma delimited text file</i>	
<a href="#">TableSchema::__get()</a>	0
<i>returns an object property</i>	
<a href="#">Tester::getAllUnitTestsCases()</a>	0
<i>return all unit test cases</i>	
<a href="#">Tester::getFunctionalTestCase()</a>	0
<i>return functional test case</i>	
<a href="#">Tester::getUnitTestCase()</a>	0
<i>return unit test case</i>	
<a href="#">Tester::loadTestConfig()</a>	0
<i>run all test cases</i>	
<a href="#">Tester::getAllFunctionalTestsCases()</a>	0
<i>return all functional test cases</i>	

<a href="#">Tester</a>	0
<i>Provides the base functionality of the application. This class receives command line data and executes the application</i>	
<a href="#">trap()</a>	0
<i>begin trapping memory</i>	
<a href="#">typemismatchexception.class.php</a>	0
<a href="#">tester.class.php</a>	0
<a href="#">Tester::runAllTestCases()</a>	0
<i>run all test cases</i>	
<a href="#">Tester::runFunctionalTestCase()</a>	0
<i>run functional test case</i>	
<a href="#">transactionexception.class.php</a>	0
<a href="#">TableSchema</a>	0
<i>Represents a table schema</i>	
<a href="#">TableSchema::seek()</a>	0
<i>seek</i>	
<a href="#">transactionbase.class.php</a>	0
<a href="#">textdataadapter.class.php</a>	0
<a href="#">Tester::runUnitTestCase()</a>	0
<i>run unit test case</i>	
<a href="#">TypeMismatchException</a>	0
<i>Represents a Type Mismatch Exception</i>	
<a href="#">tableschema.class.php</a>	0
<a href="#">TransactionBase::beginTransaction()</a>	0
<i>Begins a transaction</i>	
<a href="#">TransactionBase::commit()</a>	0
<i>Implements a commit</i>	
<a href="#">TestCaseBase::getFixtureVariables()</a>	0
<i>returns an associative array of fixture variables used for dynamically inserting values inside fixtures</i>	
<a href="#">TestCaseBase::loadCSVFixture()</a>	0
<i>load csv fixture</i>	
<a href="#">TestCaseBase::loadFixtures()</a>	0
<i>load fixtures</i>	
<a href="#">TestCaseBase::loadSQLFixture()</a>	0
<i>load sql fixture</i>	
<a href="#">TestCaseBase::cleanup()</a>	0
<i>called after every test</i>	
<a href="#">TestCaseBase::\$fixtures</a>	0
<i>fixtures to load</i>	
<a href="#">testcasebase.class.php</a>	0
<a href="#">testreporter.class.php</a>	0
<a href="#">TestCaseBase</a>	0
<i>Provides base functionality for the TestCase</i>	
<a href="#">TestCaseBase::loadXMLFixture()</a>	0
<i>load xml fixture</i>	
<a href="#">TestCaseBase::prepare()</a>	0
<i>called before every test</i>	
<a href="#">Timer::elapsed()</a>	0
<i>return elapsed time from start, if timer has been stopped then time will be calculated based on stop time</i>	
<a href="#">Timer::pause()</a>	0
<i>pause the timer</i>	

<a href="#">Timer::reset()</a>	0
<i>reset timer</i>	
<a href="#">Timer</a>	0
<i>Provides basic timer functionality</i>	
<a href="#">timer.class.php</a>	0
<a href="#">TestCaseBase::run()</a>	0
<i>run report</i>	
<a href="#">TestReporter</a>	0
<i>Provides basic reporting for test cases</i>	
<a href="#">TestReporter::setElapsedTime()</a>	0
<i>set elapsed time</i>	
<a href="#">TableCredential::generateHash()</a>	0
<i>generate password hash</i>	
<a href="#">TableCredential::comparePassword()</a>	0
<i>compare passwords, return true on success</i>	
<a href="#">translatorexception.class.php</a>	0
<a href="#">TranslatorBase</a>	0
<i>Represents a message Translator</i>	
<a href="#">TranslatorBase::get()</a>	0
<i>get</i>	
<a href="#">TranslatorBase::setCharset()</a>	0
<i>set character set</i>	
<a href="#">translatorbase.class.php</a>	0
<a href="#">TransactionException</a>	0
<i>Represents a Transaction Exception</i>	
<a href="#">TransactionBase::commitTransaction()</a>	0
<i>Implements a commit</i>	
<a href="#">TransactionBase::rollback()</a>	0
<i>Implements a rollback</i>	
<a href="#">TransactionBase::rollbackTransaction()</a>	0
<i>Implements a rollback</i>	
<a href="#">TranslatorBase::setLang()</a>	0
<i>set language</i>	
<a href="#">TranslatorException</a>	0
<i>Represents a Translator Exception</i>	
<a href="#">tablecredential.class.php</a>	0
<a href="#">TableCredential</a>	0
<i>Provides application wide authentication</i>	
<a href="#">TableCredential::authenticate()</a>	0
<i>authenticates password based on the credential</i>	
<a href="#">Test::handleException()</a>	0
<i>event triggered by an uncaught Exception thrown in the application, can be overridden to provide error handling.</i>	
<a href="#">Test::handleError()</a>	0
<i>event triggered by an error in the application, can be overridden to provide error handling.</i>	
<a href="#">test.class.php</a>	0
<a href="#">Test</a>	0
<i>deplomyent script</i>	
<a href="#">Test::execute()</a>	0
<i>execute the application</i>	
<a href="#">trace()</a>	0
<i>Trace</i>	

## U

<a href="#">UniqueValidator</a>	0
<i>Provides basic validation for web controls</i>	
<a href="#">urlvalidator.class.php</a>	0
<a href="#">UniqueValidator::onLoad()</a>	0
<i>on load</i>	
<a href="#">UniqueValidator::validate()</a>	0
<i>sets the controlId and prepares the control attributes</i>	
<a href="#">URLValidator::onLoad()</a>	0
<i>on load</i>	
<a href="#">URLValidator</a>	0
<i>Provides basic validation for web controls</i>	
<a href="#">uniquevalidator.class.php</a>	0
<a href="#">UnitTestCaseBase::tearDown()</a>	0
<i>clean test module</i>	
<a href="#">UserCredential::authenticate()</a>	0
<i>authenticates password based on the credential</i>	
<a href="#">UserCredential</a>	0
<i>Provides application wide authentication</i>	
<a href="#">UserCredential::comparePassword()</a>	0
<i>compare passwords, return true on success</i>	
<a href="#">unittestcasebase.class.php</a>	0
<a href="#">UnitTestCaseBase::setUp()</a>	0
<i>setup test module</i>	
<a href="#">UnitTestCaseBase</a>	0
<i>Provides base functionality for the UnitTestCase</i>	
<a href="#">usercredential.class.php</a>	0

## V

<a href="#">View::assign()</a>	0
<i>assign template variables to the view</i>	
<a href="#">View::fetch()</a>	0
<i>returns an html string</i>	
<a href="#">View::fetchTemplate()</a>	0
<i>return template</i>	
<a href="#">View::addHeader()</a>	0
<i>add header</i>	
<a href="#">View::add()</a>	0
<i>adds child control to collection</i>	
<a href="#">View::\$contentType</a>	0
<i>specifies content-Type, default is text/plain</i>	
<a href="#">View::\$template</a>	0
<i>specifies view template</i>	
<a href="#">View::getDomObject()</a>	0
<i>returns a DomObject for rendering</i>	
<a href="#">View::onInit()</a>	0
<i>Event called when control is initiated</i>	
<a href="#">View::sendHeaders()</a>	0
<i>send headers</i>	
<a href="#">View::setData()</a>	0



set data	0
<a href="#">View::setMaster()</a>	0
assign master view to this view	
<a href="#">View::render()</a>	0
This method will render the control to the html canvass	
<a href="#">View::onRequest()</a>	0
Event called when request is processed	
<a href="#">View::onLoad()</a>	0
Event called when page and all controls are loaded	
<a href="#">View::onPost()</a>	0
handle post events	
<a href="#">View::\$charset</a>	0
specifies charset, default is utf-8	
<a href="#">View</a>	0
Represents a Page Control	
<a href="#">ValidatorBase</a>	0
Provides basic validation for web controls	
<a href="#">ValidatorBase::\$controlToValidate</a>	0
control to validate	
<a href="#">ValidatorBase::\$errorMessage</a>	0
error message	
<a href="#">validatorcollection.class.php</a>	0
<a href="#">validatorbase.class.php</a>	0
<a href="#">ValueCollection</a>	0
Represents a collection of values	
<a href="#">ValueCollection::itemAt()</a>	0
return value at a specified index	
<a href="#">ValidatorBase::onLoad()</a>	0
on load	
<a href="#">ValidatorBase::setControlToValidate()</a>	0
set control to validate	
<a href="#">ValidatorCollection::contains()</a>	0
returns true if item array contains item	
<a href="#">view.class.php</a>	0
<a href="#">ValidatorCollection::add()</a>	0
add Validator to Collection	
<a href="#">ValidatorCollection</a>	0
Represents a Collection of Validator objects	
<a href="#">ValidatorBase::setErrorMessage()</a>	0
set error message	
<a href="#">ValidatorBase::validate()</a>	0
validates the control	
<a href="#">valuecollection.class.php</a>	0

## W

<a href="#">WebControlBase::onLoad()</a>	0
Event called when page and all controls are loaded	
<a href="#">WebControlBase::onLoadViewState()</a>	0
Event called when view state is loaded	
<a href="#">WebControlBase::onPost()</a>	0

<i>handle post events</i>	0
<a href="#"><u>WebControlBase::onInit()</u></a>	0
<i>Event called when control is initiated</i>	
<a href="#"><u>WebControlBase::onDataBind()</u></a>	0
<i>Event called when DataBind is called</i>	
<a href="#"><u>WebControlBase::load()</u></a>	0
<i>called when all controls are loaded</i>	
<a href="#"><u>WebControlBase::loadViewState()</u></a>	0
<i>read view state from session</i>	
<a href="#"><u>WebControlBase::onPreRender()</u></a>	0
<i>Event called when control is ready for rendering</i>	
<a href="#"><u>WebControlBase::onRequest()</u></a>	0
<i>Event called when request is processed</i>	
<a href="#"><u>WebControlBase::removeControl()</u></a>	0
<i>remove a child control</i>	
<a href="#"><u>WebControlBase::render()</u></a>	0
<i>This method will render the control to the html canvass</i>	
<a href="#"><u>WebControlBase::refreshDataSource()</u></a>	0
<i>refresh data source</i>	
<a href="#"><u>WebControlBase::preRender()</u></a>	0
<i>called before control is ready for rendering</i>	
<a href="#"><u>WebControlBase::onSaveViewState()</u></a>	0
<i>Event called when view state is written</i>	
<a href="#"><u>WebControlBase::onUpdateAjax()</u></a>	0
<i>Event called on ajax callback</i>	
<a href="#"><u>WebControlBase::isControl()</u></a>	0
<i>compares control id, returns true if control id matches</i>	
<a href="#"><u>WebControlBase::init()</u></a>	0
<i>called when control is initiated</i>	
<a href="#"><u>WebControlBase::getControl()</u></a>	0
<i>returns reference to child control</i>	
<a href="#"><u>WebControlBase::getDomObject()</u></a>	0
<i>returns a DomObject representing control</i>	
<a href="#"><u>WebControlBase::findControl()</u></a>	0
<i>returns WebControl if control is found in Collection</i>	
<a href="#"><u>WebControlBase::fetch()</u></a>	0
<i>returns an html string</i>	
<a href="#"><u>WebControlBase::createDomObject()</u></a>	0
<i>returns the default DomObject</i>	
<a href="#"><u>WebControlBase::escape()</u></a>	0
<i>escape</i>	
<a href="#"><u>WebControlBase::getHTMLControlId()</u></a>	0
<i>creates controlId string for html</i>	
<a href="#"><u>WebControlBase::getHTMLControlIdString()</u></a>	0
<i>creates controlId string for html</i>	
<a href="#"><u>WebControlBase::handlePostEvents()</u></a>	0
<i>handle post events</i>	
<a href="#"><u>WebControlBase::hasControl()</u></a>	0
<i>returns Webtrue if control is found in Collection</i>	
<a href="#"><u>WebControlBase::getRequestData()</u></a>	0
<i>creates string of all vars currently in the request object seperated by an '&amp;' used to preserve application state</i>	

<a href="#">WebControlBase::getQueryString()</a>	0
<i>creates string of all vars currently in the request object seperated by an '&amp;'</i> <i>used to preserve application state</i>	
<a href="#">WebControlBase::getParentByType()</a>	0
<i>returns 1st parent object by type</i>	
<a href="#">WebControlBase::getParentString()</a>	0
<i>returns string of parent id's seperated by .</i>	
<a href="#">WebControlBase::replaceControl()</a>	0
<i>replace a child control</i>	
<a href="#">WebControlBase::requestProcessor()</a>	0
<i>process the HTTP request array</i>	
<a href="#">WebServiceBase::\$reserved_methods</a>	0
<i>array of reserved methods</i>	
<a href="#">WebServiceBase::\$view</a>	0
<i>contains the View object</i>	
<a href="#">WebServiceBase::\$remoteProcedures</a>	0
<i>Contains an array of RPC method calls</i>	
<a href="#">WebServiceBase</a>	0
<i>This class handles all remote procedure calls for a web service</i>	
<a href="#">WebControlPreRenderEvent</a>	0
<i>Provides event handling</i>	
<a href="#">WebControlPreRenderEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">WebServiceBase::configure()</a>	0
<i>configure the server</i>	
<a href="#">WebServiceBase::getView()</a>	0
<i>return view component for rendering</i>	
<a href="#">WebServiceMethod::setFunction()</a>	0
<i>set function</i>	
<a href="#">WebServiceMethod::setParameters()</a>	0
<i>set parameters</i>	
<a href="#">WebServiceMethod::getParameters()</a>	0
<i>return all parameters</i>	
<a href="#">WebServiceMethod::getFunction()</a>	0
<i>return the function callback</i>	
<a href="#">WebServiceBase::handle()</a>	0
<i>this method will handle the web service request</i>	
<a href="#">WebServiceMethod</a>	0
<i>This class represents a remote procedure call</i>	
<a href="#">WebControlLoadEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">WebControlLoadEvent</a>	0
<i>Provides event handling</i>	
<a href="#">WebControlCollection::add()</a>	0
<i>add WebControl to Collection</i>	
<a href="#">WebControlCollection::contains()</a>	0
<i>returns true if array item is found</i>	
<a href="#">WebControlCollection</a>	0
<i>Represents a Collection of WebControl objects</i>	
<a href="#">WebControlBase::updateAjax()</a>	0
<i>updates the ui control asynchronously</i>	
<a href="#">WebControlBase::saveViewState()</a>	0
<i>write view state to session</i>	

<a href="#">WebControlBase::setParent()</a>	0
<i>set parent control</i>	
<a href="#">WebControlCollection::indexOf()</a>	0
<i>returns index if value is found in collection</i>	
<a href="#">WebControlCollection::itemAt()</a>	0
<i>return WebControl at a specified index</i>	
<a href="#">WebControlInitEvent</a>	0
<i>Provides event handling</i>	
<a href="#">WebControlInitEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">WebControlCreateEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">WebControlCreateEvent</a>	0
<i>Provides event handling</i>	
<a href="#">WebControlCollection::remove()</a>	0
<i>remove item from collection</i>	
<a href="#">WebControlCollection::replace()</a>	0
<i>replace a WebControl in a Collection</i>	
<a href="#">WebControlBase::bind()</a>	0
<i>bind a data source to control</i>	
<a href="#">WebControlBase::addControl()</a>	0
<i>add a child control to collection</i>	
<a href="#">webapplicationsavestateeventhandler.class.php</a>	0
<a href="#">WebApplicationHandleRequestEvent</a>	0
<i>Provides event handling</i>	
<a href="#">webapplicationsavestateevent.class.php</a>	0
<a href="#">webapplicationloadstateeventhandler.class.php</a>	0
<a href="#">webapplicationhandlerequesteventhandler.class.php</a>	0
<a href="#">webapplicationloadstateevent.class.php</a>	0
<a href="#">WebApplicationHandleRequestEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">WebApplicationLoadStateEvent</a>	0
<i>Provides event handling</i>	
<a href="#">WebServiceAuthentication</a>	0
<i>Provides application wide authentication using Basic HTTP Headers</i>	
<a href="#">WebServiceAuthentication::authenticated()</a>	0
<i>returns true if user authenticated</i>	
<a href="#">webserviceauthentication.class.php</a>	0
<a href="#">WebApplicationSaveStateEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">WebApplicationLoadStateEventHandler</a>	0
<i>Provides event handling</i>	
<a href="#">WebApplicationSaveStateEvent</a>	0
<i>Provides event handling</i>	
<a href="#">webapplicationhandlerequestevent.class.php</a>	0
<a href="#">WebApplicationBase::trace()</a>	0
<i>Trace a variable</i>	
<a href="#">WebApplicationBase::getPageURI()</a>	0
<i>returns a URI based on the requested page and parameters</i>	
<a href="#">WebApplicationBase::getRequestHandler()</a>	0
<i>return the requestHandler</i>	
<a href="#">WebApplicationBase::getAllControllers()</a>	0
<i>return all controllers</i>	

<a href="#">WebApplicationBase::execute()</a>	0
execute the application	
<a href="#">WebApplicationBase</a>	0
This class represents the base web application. This class receives HTTP request input and delegates the request to a Controller. Once the Controller has handled the request, this class will regain control and render the appropriate View selected by the Controller unless there are additional actions to perform.	
<a href="#">WebApplicationBase::clearForwardPage()</a>	0
This method clear all forward URI's	
<a href="#">WebApplicationBase::getSession()</a>	0
returns the common translator object (overridden this method in the application class)	
<a href="#">WebApplicationBase::handleError()</a>	0
event triggered by an error in the application, can be overridden to provide error handling.	
<a href="#">WebApplicationBase::sendHTTPError()</a>	0
send HTTP status message to client	
<a href="#">WebApplicationBase::setForwardPage()</a>	0
This method sets the next URI once the current controller is finished executing, the servlet will re-request the page with the requested action. This method allows you to replace post data with keep friendly urls that can be bookmarked.	
<a href="#">WebApplicationBase::saveApplicationState()</a>	0
This method stores an instance of the previous action and the servlet messages. These can then	
be used by the servlet after a page request to populate the view.	
<a href="#">WebApplicationBase::requestProcessor()</a>	0
Processes HTTP Request and create actions to handle business logic. Then render the view.	
<a href="#">WebApplicationBase::handleException()</a>	0
event triggered by an uncaught Exception thrown in the application, can be overridden to provide error handling.	
<a href="#">WebApplicationBase::loadApplicationState()</a>	0
This method retrieves an instance of the previous action and the servlet messages. These can then	
be used by the servlet to populate the view.	
<a href="#">WebServiceAuthentication::authenticateSecret()</a>	0
authenticate secret	
<a href="#">WebServiceAuthentication::getAuthSecret()</a>	0
gets auth secret	
<a href="#">WebControlBase::\$ajaxCallback</a>	0
specifies the ajax callback uri (optional)	
<a href="#">WebControlBase::\$ajaxEventHandler</a>	0
specifies the name of the ajax event handler javascript function (optional)	
<a href="#">WebControlBase</a>	0
Provides a web-based server-side UI to manage the HTML that ends up on client machine by abstracting away the presentation logic and data access layer	
<a href="#">WebControlAttributeCollection::append()</a>	0
append attribute of xml node	
overwrite attribute with same name	
<a href="#">webcontrolcollection.class.php</a>	0
<a href="#">WebControlAttributeCollection</a>	0
Represents a Collection of WebControl attributes	
<a href="#">WebControlBase::\$ajaxHttpRequest</a>	0
specifies the name of the ajax HttpRequest object (optional)	
<a href="#">WebControlBase::\$attributes</a>	0
collection of arbitrary attributes (for rendering only)	

<a href="#">WebControlBase::\$events</a>	0
<i>event collection</i>	
<a href="#">WebControlBase::\$visible</a>	0
<i>determines whether the control is visible, Default is true</i>	
<a href="#">WebControlBase::\$escapeOutput</a>	0
<i>determines whether to escape the output</i>	
<a href="#">WebControlBase::\$enableViewState</a>	0
<i>specifies whether the server control persists its view state, Default is true</i>	
<a href="#">WebControlBase::\$controls</a>	0
<i>collection of child controls</i>	
<a href="#">WebControlBase::\$dataSource</a>	0
<i>instance of the datasource</i>	
<a href="#">webcontrolbase.class.php</a>	0
<a href="#">webcontrolattributecollection.class.php</a>	0
<a href="#">webcontrolcreateevent.class.php</a>	0
<a href="#">webcontrolcreateeventhandler.class.php</a>	0
<a href="#">WebServiceAuthentication::signout()</a>	0
<i>perform sign out (does not end session)</i>	
<a href="#">WebServiceAuthentication::setAuthUser()</a>	0
<i>sets auth user</i>	
<a href="#">WebServiceAuthentication::getAuthUser()</a>	0
<i>gets auth user</i>	
<a href="#">WebServiceAuthentication::isAuthUserSet()</a>	0
<i>returns true if auth is set</i>	
<a href="#">webcontrolinitevent.class.php</a>	0
<a href="#">webcontroliniteventhandler.class.php</a>	0
<a href="#">webservicebase.class.php</a>	0
<a href="#">webservicemethod.class.php</a>	0
<a href="#">webcontrolprerendereventhandler.class.php</a>	0
<a href="#">webcontrolprerenderevent.class.php</a>	0
<a href="#">webcontrolloadevent.class.php</a>	0
<a href="#">webcontrolloadeventhandler.class.php</a>	0
<a href="#">webapplicationbase.class.php</a>	0

## X

<a href="#">XMLEntity::replaceChild()</a>	0
<i>replace a reference to a child element</i>	
<a href="#">XMLEntity::setAttribute()</a>	0
<i>set entity attribute</i>	
<a href="#">XMLEntityCollection</a>	0
<i>Represents a Collection of XMLEntity objects</i>	
<a href="#">XMLEntity::removeChild()</a>	0
<i>remove a reference to a child element</i>	
<a href="#">XMLEntity::removeAttribute()</a>	0
<i>remove an attribute from an xml node</i>	
<a href="#">XMLEntity::getChildrenByName()</a>	0
<i>returns all children by name</i>	
<a href="#">XMLEntity::getEntityValue()</a>	0
<i>get formatted entity value</i>	
<a href="#">XMLEntity::getXMLString()</a>	0
<i>return XML string</i>	

<a href="#">XMLEntityCollection::add()</a>	0
<i>add XMLEntity to Collection</i>	
<a href="#">XMLEntityCollection::contains()</a>	0
<i>returns true if array item is found</i>	
<a href="#">XMLParser::\$data</a>	0
<i>array of xml data</i>	
<a href="#">XMLParser::\$error</a>	0
<i>parse error</i>	
<a href="#">XMLParser::parse()</a>	0
<i>parse xml data</i>	
<a href="#">XMLParser</a>	0
<i>Provides a way to parse xml documents into XMLEntity objects</i>	
<a href="#">XMLException</a>	0
<i>Represents an XML Exception</i>	
<a href="#">XMLEntityCollection::indexOf()</a>	0
<i>returns index if value is found in collection</i>	
<a href="#">XMLEntityCollection::itemAt()</a>	0
<i>return XMLEntity at a specified index</i>	
<a href="#">XMLEntity::getChildByName()</a>	0
<i>returns a reference to a child element by element name</i>	
<a href="#">XMLEntity::getChildByAttribute()</a>	0
<i>returns a reference to a child element by element attribute</i>	
<a href="#">XMLEntity::\$attributes</a>	0
<i>entity attributes</i>	
<a href="#">XMLEntity::\$cdata</a>	0
<i>cdata value</i>	
<a href="#">XMLEntity::\$children</a>	0
<i>entity children</i>	
<a href="#">XMLEntity</a>	0
<i>XMLEntity</i>	
<a href="#">xmlparser.class.php</a>	0
<a href="#">xmlentitycollection.class.php</a>	0
<a href="#">xmlexception.class.php</a>	0
<a href="#">XMLEntity::\$forceCloseTag</a>	0
<i>set to force close tag</i>	
<a href="#">XMLEntity::\$name</a>	0
<i>entity name</i>	
<a href="#">XMLEntity::findChildByName()</a>	0
<i>returns a reference to a child element by element name</i>	
<a href="#">XMLEntity::getAttribute()</a>	0
<i>get attribute of xml node</i>	
<a href="#">XMLEntity::getChild()</a>	0
<i>returns a reference to a child element by index</i>	
<a href="#">XMLEntity::fetch()</a>	0
<i>return XML node as string</i>	
<a href="#">XMLEntity::appendAttribute()</a>	0
<i>append attribute of xml node</i>	
<i>overwrite attribute with same name</i>	
<a href="#">XMLEntity::\$value</a>	0
<i>entity value</i>	
<a href="#">XMLEntity::addChild()</a>	0
<i>add a reference to a child element</i>	
<a href="#">xmlentity.class.php</a>	0



<u>NAME</u>	0
specifies the server name	
<u>MODELS_PATH</u>	0
specifies the default model path	
<u>PAGE_EXTENSION</u>	0
specifies the request page extension	
<u>PAGE_REQUEST_PARAMETER</u>	0
specifies the page request parameter	
<u>PLUGINS_PATH</u>	0
specifies the root relative path to the plugins folder	
<u>MODELS_NAMESPACE</u>	0
specifies the models namespace	
<u>MIGRATIONS_PATH</u>	0
specifies the root relative path to the migrations folder	
<u>LANGS_TABLENAME</u>	0
specifies the langs table name	
<u>LANGS_FILE</u>	0
specifies the path to the langs file	
<u>LIB_PATH</u>	0
specifies the root relative path to the libs folder	
<u>LOGS_TABLENAME</u>	0
specifies the logs table name	
<u>LOG_PATH</u>	0
specifies the root relative path to the log folder	
<u>PORT</u>	0
specifies the server port	
<u>PROTOCOL</u>	0
specifies the transfer protocol	
<u>TEST_ENV</u>	0
specifies the test environment name	
<u>TESTCASE_SUFFIX</u>	0
specifies the testcase suffix	
<u>UNIT_TESTS_PATH</u>	0
specifies the default unit test path	
<u>VALIDATION_TIMEOUT</u>	0
specifies the asynchronous validation timeout	
<u>VIEWS_PATH</u>	0
specifies the default view path	
<u>TEMPLATE_EXTENSION</u>	0
specifies the template extension	
<u>TEMPLATES_PATH</u>	0
specifies the default template path	
<u>QUOTE_STYLE</u>	0
specifies the quote style when escaping	
<u>ROOT</u>	0
<u>SSL_PORT</u>	0
specifies the SSL port	
<u>SYSTEM_PATH</u>	0
specifies the root relative path to the system folder	
<u>HTDOCS_PATH</u>	0
specifies the public web folder path	



<u>HOST</u>	0
<i>specifies the host name</i>	
<u>CLASS_EXTENSION</u>	0
<i>specifies the class extension</i>	
<u>CACHE_TABLENAME</u>	0
<i>specifies the cache table name</i>	
<u>CONFIG_PATH</u>	0
<i>specifies the root relative path to the configuration folder</i>	
<u>CONTROLLERS_NAMESPACE</u>	0
<i>specifies the models namespace</i>	
<u>CONTROLLERS_PATH</u>	0
<i>specifies the default controller path</i>	
<u>CACHE_PATH</u>	0
<i>specifies the root relative path to the cache folder</i>	
<u>BUILD_PATH</u>	0
<i>specifies the root relative path to the cache folder</i>	
<u>APP_PATH</u>	0
<i>specifies the root relative path to the application folder</i>	
<u>APP_CONF_FILENAME</u>	0
<i>specifies the environment parameter</i>	
<u>APP_URI</u>	0
<i>specifies the application uri</i>	
<u>ASSETS_URI</u>	0
<i>specifies the URI to the assets folder</i>	
<u>ASYNC_REQUEST_PARAMETER</u>	0
<i>specifies the async request parameter</i>	
<u>CONTROLLER_EXTENSION</u>	0
<i>specifies the controller extension</i>	
<u>CONTROLLER_TESTCASE_SUFFIX</u>	0
<i>specifies the controller testcase suffix</i>	
<u>ENV_PATH</u>	0
<i>specifies the root relative path to the configuration folder</i>	
<u>ENV_PARAMETER</u>	0
<i>specifies the environment parameter</i>	
<u>ERROR_LIMIT</u>	0
<i>specifies the number of warnings to dump</i>	
<u>FIXTURES_PATH</u>	0
<i>specifies the default fixtures path</i>	
<u>FUNCTIONAL_TESTS_PATH</u>	0
<i>specifies the default functional test path</i>	
<u>ENV_CONF_FILENAME</u>	0
<i>specifies the environment parameter</i>	
<u>DEV_ENV</u>	0
<i>specifies the dev environment name</i>	
<u>DB_SCHEMA_VERSION_TABLENAME</u>	0
<i>specifies the db_schema_version table name</i>	
<u>DEFAULT_TIMEZONE</u>	0
<i>specifies the default timezone</i>	
<u>DEPLOYMENT_EXTENSION</u>	0
<i>specifies the controller extension</i>	
<u>DEPLOY_PATH</u>	0
<i>specifies the root relative path to the deployment folder</i>	
<u>ACTIVERECORD_AUTO_MAP</u>	0

*specifies the number of warnings to dump*