

**NAME**

MQdb::DBObject – DESCRIPTION of Object

**SYNOPSIS**

Root class for all objects in MappedQuery toolkit

**DESCRIPTION**

Root object for toolkit and all derived subclasses. All objects in the MappedQuery structure are designed to be persisted in a database. Here database is a more broad term and can be considered any object persistence systems. Currently the toolkit works with SQL based systems but object databases or custom storage engines are possible. Provides base common methods used by all objects.

**AUTHOR**

Contact Jessica Severin: jessica DOT severin AT gmail DOT com

**LICENSE**

```
* Software License Agreement (BSD License)
* MappedQueryDB [MQdb] toolkit
* copyright (c) 2006-2009 Jessica Severin
* All rights reserved.
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:
*     * Redistributions of source code must retain the above copyright
*       notice, this list of conditions and the following disclaimer.
*     * Redistributions in binary form must reproduce the above copyright
*       notice, this list of conditions and the following disclaimer in the
*       documentation and/or other materials provided with the distribution.
*     * Neither the name of Jessica Severin nor the
*       names of its contributors may be used to endorse or promote products
*       derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS 'AS IS' AND ANY
* EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS BE LIABLE FOR ANY
* DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

**APPENDIX**

The rest of the documentation details each of the object methods. Internal methods are usually preceded with a \_

**new**

```
Description: instance creation method
Returntype : instance of this Class (subclass)
Exceptions : none
```

**init**

```
Description: initialization method which subclasses can extend
Returntype : $self
Exceptions : subclass dependent
```

**copy**

```
Description: Shallow copy which copies all base attributes of object
              to new instance of same class
Returntype : same as calling instance
Exceptions : subclass dependent
```



**class**

Description: fixed string symbol for this class. Must be implemented for each subclass and each subclass within toolkit should return a unique name. used by global methods.

Returntype : string

Exceptions : error if subclass does not redefine

**database**

Description: the MQdb::Database where this object is permanently persisted to. Here database is any object persistence system.

Returntype : MQdb::Database

Exceptions : die if invalid setter value type is provided

**primary\_id**

Description: the unique identifier for this object within database.

Returntype : scalar or UNDEF

Exceptions : none

**id**

Description: the unique identifier for this object within database. Returns empty string if not persisted.

Returntype : scalar or ''

Exceptions : none

**db\_id**

Description: the worldwide unique identifier for this object. A URL-like combination of database, class, and id

Returntype : string or undef if database is not defined

Exceptions : none

**display\_desc**

Description: general purpose debugging method that returns a nice human readable description of the object instance contents. Each subclass should implement and return a nice string.

Returntype : string scalar

Exceptions : none

**display\_info**

Description: convenience method which prints the display\_desc string with a carriage return to STDOUT. useful for debugging.

Returntype : none

Exceptions : none

**xml**

Description: every object in system should be persistable in XML format. returns an XML description of the object and all child objects. Each subclass must implement and return a proper XML string. Best if one implements xml\_start() and xml\_end() and use here.

Returntype : string scalar

Exceptions : none

Default : default is a simple xml\_start + xml\_end

**xml\_start**

Description: every object in system should be persistable in XML format. returns an XML description of the object and all child objects. Each subclass should OVERRIDE this method and return a proper XML. xml\_start is the primary XML start tag

Example : return sprintf("<feature id='%d' name='%d' ..... >", \$id, \$name...

Returntype : string scalar

Exceptions : none

**xml\_end**

Description: every object in system should be persistable in XML format.  
returns an XML description of the object and all child objects.  
Each subclass should OVERRIDE this method and return a proper XML  
xml\_end is the primary XML end tag

Example : return "</feature>";

Returntype : string scalar

Exceptions : none

### **simple\_xml**

Description: short hand for xml\_start() . xml\_end()

Can be used when only the primary XML start tag and attributes are

No need to override if xml\_start() and xml\_end() are implemented

Returntype : string scalar

Exceptions : none

