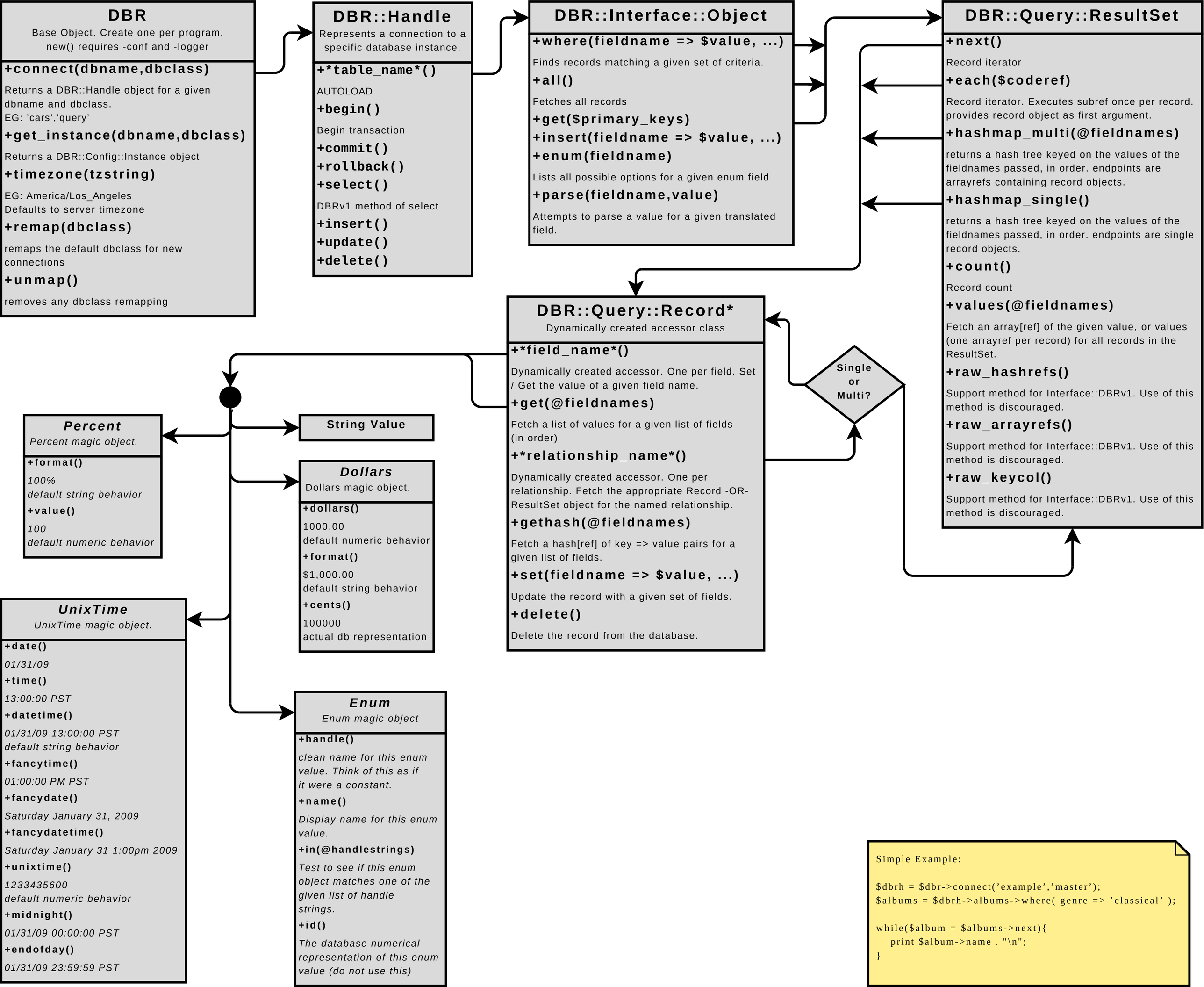
DBR
Base Object. Create one per program.
new() requires -conf and -logger
+connect(dbname,dbclass)
Returns a DBR::Handle object for a given dbname and dbclass.
EG: 'cars','query'
+get_instance(dbname,dbclass)
Returns a DBR::Config::Instance object
+timezone(tzstring)
EG: America/Los_Angeles
Defaults to server timezone
+remap(dbclass)
remaps the default dbclass for new connections
+unmap()
removes any dbclass remapping
DBR::Handle
Represents a connection to a specific database instance.
+*table_name*()
AUTOLOAD
+begin()
Begin transaction
+commit()
+rollback()
+select()
DBRv1 method of select
+insert()
+update()
+delete()
DBR::Interface::Object
+where(fieldname => $value, ...)
Finds records matching a given set of criteria.
+all()
Fetches all records
+get($primary_keys)
+insert(fieldname => $value, ...)
+enum(fieldname)
Lists all possible options for a given enum field
+parse(fieldname,value)
Attempts to parse a value for a given translated field.
DBR::Query::ResultSet
+next()
Record iterator
+each($coderef)
Record iterator. Executes subref once per record. provides record object as first argument.
+hashmap_multi(@fieldnames)
returns a hash tree keyed on the values of the fieldnames passed, in order. endpoints are arrayrefs containing record objects.
+hashmap_single()
returns a hash tree keyed on the values of the fieldnames passed, in order. endpoints are single record objects.
+count()
Record count
+values(@fieldnames)
Fetch an array[ref] of the given value, or values (one arrayref per record) for all records in the ResultSet.
+raw_hashrefs()
Support method for Interface::DBRv1. Use of this method is discouraged.
+raw_arrayrefs()
Support method for Interface::DBRv1. Use of this method is discouraged.
+raw_keycol()
Support method for Interface::DBRv1. Use of this method is discouraged.
DBR::Query::Record*
Dynamically created accessor class
+*field_name*()
Dynamically created accessor. One per field. Set / Get the value of a given field name.
+get(@fieldnames)
Fetch a list of values for a given list of fields (in order)
+*relationship_name*()
Dynamically created accessor. One per relationship. Fetch the appropriate Record -OR- ResultSet object for the named relationship.
+gethash(@fieldnames)
Fetch a hash[ref] of key => value pairs for a given list of fields.
+set(fieldname => $value, ...)
Update the record with a given set of fields.
+delete()
Delete the record from the database.
Single or Multi?
Percent
Percent magic object.
+format()
100%
default string behavior
+value()
100
default numeric behavior
String Value
Dollars
Dollars magic object.
+dollars()
1000.00
default numeric behavior
+format()
$1,000.00
default string behavior
+cents()
100000
actual db representation
UnixTime
UnixTime magic object.
+date()
01/31/09
+time()
13:00:00 PST
+datetime()
01/31/09 13:00:00 PST
default string behavior
+fancytime()
01:00:00 PM PST
+fancydate()
Saturday January 31, 2009
+fancydatetime()
Saturday January 31 1:00pm 2009
+unixtime()
1233435600
default numeric behavior
+midnight()
01/31/09 00:00:00 PST
+endofday()
01/31/09 23:59:59 PST
Enum
Enum magic object
+handle()
clean name for this enum value. Think of this as if it were a constant.
+name()
Display name for this enum value.
+in(@handlestrings)
Test to see if this enum object matches one of the given list of handle strings.
+id()
The database numerical representation of this enum value (do not use this)
Simple Example:
$dbrh = $dbr->connect('example','master');
$albums = $dbrh->albums->where(genre => 'classical');
while($album = $albums->next){
print $album->name . "\n";
}