

dbix概要设计文档

（仅供内部使用）

|  |  |  |
| --- | --- | --- |
| 文 件 编 号： | |  |
| 版 本 号： | |  |
| 实 施 日 期： | |  |
|  | | |
| 编 制： | 胡志云 | |
| 审 核： |  | |
| 会 签： |  | |
|  |  | |
|  |  | |
| 批 准： |  | |

**修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本号 | 描述 | 作者 |
| 20120502 | 0.1 | 初稿 | 胡志云 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1 引言 4](#_Toc338855076)

[1.1 编写目的 4](#_Toc338855077)

[1.2 预期读者和阅读建议 4](#_Toc338855078)

[1.3 参考资料 4](#_Toc338855079)

[1.4 术语与缩略语 4](#_Toc338855080)

[2 设计概述 4](#_Toc338855081)

[2.1 限制和约束 4](#_Toc338855082)

[2.2 设计原则和设计要求 5](#_Toc338855083)

[3 功能列表 5](#_Toc338855084)

[4 软件架构 6](#_Toc338855085)

[4.1 总体架构 6](#_Toc338855086)

[4.2 线程设计 7](#_Toc338855087)

[4.3 核心类图 7](#_Toc338855088)

[5 部件功能 8](#_Toc338855089)

[5.1 线程池 8](#_Toc338855090)

[5.2 连接池 8](#_Toc338855091)

[5.3 实体类封装 8](#_Toc338855092)

[5.4 命令类封装 8](#_Toc338855093)

[5.5 DBI层封装 8](#_Toc338855094)

[6 编程接口 8](#_Toc338855095)

[7 流程设计 9](#_Toc338855096)

[7.1 框架的初始化 9](#_Toc338855097)

[7.2 框架的反初始化 9](#_Toc338855098)

[7.3 操作项目的投递与处理 9](#_Toc338855099)

[8 难点及处理 9](#_Toc338855100)

[9 遗留问题 9](#_Toc338855101)

# 引言

## 编写目的

本文主要介绍了dbix框架的功能，实现原理，使初步介入的开发人员能够较快对dbix有整体认识与了解，也方便后续版本的优化与调整。

## 预期读者和阅读建议

项目经理和开发人员。

## 参考资料

1. 参考了黄至春提供的功能类，类型无关类和条件查询类
2. Hiberate相关接口
3. 平台1.0中UAS和TAS的使用场景，以及2.0中TAS提炼的接口

## 术语与缩略语

无。

# 设计概述

dbix的全称是database interface x, x表示类似扩展、加强等含义。dbix是一个数据库编程框架，采用“异步请求+结果回调返回”的编程模型。dbix实现数据库的对象化，屏蔽数据库差异，数据库连接、SQL执行、数据获取等操作。

dbix对数据库中的对象进行抽象。数据库中的表和视图抽象成实体类，数据库中的存储过程抽象成命令类；实体类和命令类支持派生自定义实现。用户使用时生成相应的对象，然后调用操作函数，最后dbix会将结果利用回调函数的方式返回。

dbix对简单的数据库操作可以自动生成相应的SQL语句，此功能可以减少开发工作量和降低数据库的编程门槛。用户也可以派生实体类或者命令类，从而达到自己手写特殊要求的SQL语句的目的。

## 限制和约束

1. dbix对操作系统和数据库的支持分别如下：

操作系统：windows, linux(比如CentOS、Redhat)；

数据库：mysql, sybase, oracle。

1. dbix不保证操作的时序性，也即先投递的请求不一定先执行。

## 设计原则和设计要求

无。

# 功能列表

总体上要求能够异步地以对象化方式访问数据库中的数据和执行数据库中的命令，另外简单的SQL语句要求能够自动生成。下面对需求进行细化。

基本需求细化：

1. 封装DB线程池，支持线程数动态变化
2. 屏蔽DB连接细节, 连接断开自动重连
3. 封装数据库差异，SQL语句非标准部分：分页（?），存储过程执行，语句分隔符，查询通配符
4. 提供辅助接口，SQL字段编码，SQL语句拼接辅助方法和结果获取辅助方法
5. SQL语句的条件控制要求满足所有可能用到的SQL条件

扩展需求细化：

1. 封装数据库操作方式为“异步请求+回调返回结果”的工作方式
2. 简单的表操作SQL语句要求能够自动生成
3. 不同对象之间复用得方便，以处理一个对象同时依赖多个对象的场景
4. 支持运行过程中连接的直接管理，用于可能存在的操作多个数据库或者动态切换连接
5. 支持用户自定义实现逻辑

原始讨论和思考记录，附件——。

# 软件架构

## 总体架构



图 3‑2 dbix架构图

用户构造数据库操作请求，然后投递到任务队列，马上返回。dbix框架的线程池中的线程不断地从任务队列中提取任务，利用线程拥有的连接，在它上面执行任务；线程执行完任务时利用回调函数将结果返回。

为了与数据库进行通信，dbix需要调用我司的dbi库；dbi会屏蔽不同数据库之间通信层面的差异性。

## 线程设计

dbix框架只设计了线程池中的一组线程。

## 核心类图





# 部件功能

下面简单介绍一下dbix内部各个部件的功能。

## 线程池

采用类似于epoll模型的方式，不断地取操作对象队列中的操作对象执行然后利用回调返回结果。

## 连接池

统一管理所有的连接，实现连接的重复利用。

## 实体类封装

抽象数据库中的表或视图，内置支持简单的添加、删除、修改和查询操作；用户也可以派生子类，然后自定义前面的四种操作。

## 命令类封装

抽象数据库中的存储过程，支持存储过程的调用命令生成；用户也可以派生子类，然后自定义调用命令的生成和提取结果等。

## DBI层封装

封装DBI调用，以及SQL语法的不同数据库之间的差异性。

# 编程接口

采用“异步请求+回调返回结果”的编程模型。

接口参考dbi.h头文件，使用方式参考dbix的demo工程。

伪代码demo, 附件­——。

# 流程设计

## 框架的初始化

## 框架的反初始化

## 操作项目的投递与处理

# 难点及处理

1. 表数据完整性，保证尽量做在实体层，如此可以减少交互次数，同时也有利于统一保证表数据的完整性。
2. 过滤条件类，只做到自身属性作为过滤条件的场景，因为只有查询操作最有可能用到。复杂的过滤条件（包括联表与嵌套子查询等），用到的时候要求采用自定义操作类(自定义实体类或者命令类)的方式，自定义操作类上面再额外设计限制条件。另外，只限制自身属性作为过滤条件的时候，在添加条件时我们可以做校验，判断此条件的合法性。
3. NULL条件不直接支持，数据列最好不要出现判断NULL的场景，可能导致索引失效等问题。

# 遗留问题