

# 设计文档

---

本文档描述了一个基于Qt的桌面日历程序的实现架构。

本程序的整体架构分为使用QML编写的界面框架和控制部分，和使用C++语言编写的三个辅助事务管理类。QML和C++组件之间使用Qt Quick的QML-C++接口和信号-槽机制进行连接。

本程序主要由以下的组件组成：

## ApplicationManager 类

---

ApplicationManager 类负责处理与应用程序有关的接口，包含以下一些主要成员：

- 接口 addShortcut 用于添加一个全局快捷键
- 接口 loadConfig 用于读取设置文件 config.json
- 接口 saveConfig 用于写入设置文件 config.json
- 接口 changeLangauge 用于更换程序的语言
- 信号 shortcutTriggered 触发于快捷键触发时。

ApplicationManager 使用第三方库 QGlobalShortcut 管理全局快捷键。

(<https://github.com/mitei/qglobalshortcut>)

## AttachmentManager 类

---

AttachmentManager 类实现对事件的附加文件的管理，包含以下一些主要成员：

- 接口 AttachFile 接受日期和文件的URL作为参数，复制给定的文件并在数据库中新建一条相关的事件。
- 接口 getFileURI 接受日期和文件的SHA256指纹作为参数，返回文件的存储位置的URL。
- 接口 removeFile 根据给定的SHA256指纹删除文件。

AttachmentManager 将文件自动存储在工作目录下storage文件夹内以文件SHA256指纹前八位命名的文件夹内。

## EventManager 类

---

EventManager 类实现对事件的管理，包含以下一些主要成员：

- 私有函数 event\_react\_to 处理事件是否在某一天发生的实际逻辑。
- 接口 eventsForDate 接受日期作为参数，查询得到所有该天的事件并包装成为 JSON 格式返回。

- 接口 `modify` 接受 `JSON` 格式包装的事件信息和操作类型作为参数，根据规定的操作类型修改数据库内相关的内容，并处理相关的附件。
- 接口 `dbPath`、`importDatabase` 处理与导入、导出数据库有关的功能。

`EventManager` 使用SQLite关系数据库进行事件数据的管理，数据库自动储存于工作目录下的 `events.db` 文件。

数据库的字段包括 `name` , `type` , `mask` , `startDate` , `startTime` , `endDate` , `endTime` , `color` 、分别存储事件的名称、类型、mask、开始日期和开始时间、结束日期和结束时间、颜色。其中，当类型为每周或每月重复时，`mask`用于存储与重复规则的信息；当类型为文件时，`mask`用于存储文件的SA256指纹；当类型为单次事件，`mask`无意义。

`EventManager` 与程序的QML部分使用JSON格式进行数据交换，JSON格式的数组内保存的字段包括 `id` , `name` , `type` , `mask` , `startDate` , `endDate` 。其中 `id` 保存事件在数据库中的唯一ID，`startDate` 、`endDate` 使用字符串表示的整数表示开始结束的日期时间以`msecFromEpoch`形式保存的值，其他与数据库保存的方式相同。

## main.qml

---

`main.qml` 描述了程序的窗口，对设置文件进行载入，并设置了四个按钮用于编辑程序的属性和进行数据的导入导出。`main.qml` 使用一个 `Loader` 载入 `PageMonth.qml` 。

## DayGrid.qml

---

`DayGrid` 组件描绘了日历中每个日期格的结构，对日期格中的事件进行绘制并处理对于日期格的鼠标事件。

`DayGrid` 的数据模型属性绑定为日期和事件列表。`Daygrid` 组件发射的信号包括 `gridClicked` , `eventClicked` , `eventChanged` , 分别表示各自、事件的点击事件和拖放事件。

## EventEditingPopup.qml

---

`EventEditingPopup` 组件描绘了添加/修改事件的对话框的结构。`EventEditingPopup` 描述了对话框内的元素和之间的关系，并处理事件数据的编辑。`EventEditingPopup` 通过 `event` 的 `type` 对显示的条目进行筛选。

`EventEditingPopup` 的数据模型属性绑定为受编辑的事件和进行的编辑事件。`EventEditingPopup` 的处理程序主要绑定于 `accepted` 信号。

## PageMonth.qml

---

`PageMonth.qml` 描述了程序的月历页的主界面。`PageMonth` 的数据模型绑定为选中的日期，在日期更改或之星对数据库的修改后对月历页面进行重绘。

## Utils.js 库

---

`Utils.js` 库是所有QML组件的公用库，负责一些辅助函数和设置的管理。

程序的QML组件使用Qt Quick Controls 2和qml material (<https://github.com/papyros/qml-material>)。