Messaging API 文档

Messaging

Qt Messaging API 使我们可以处理信息（短信，彩信，邮件等），搜索信息，排列信息，发送信息，提取信息数据，可以调用平台对应的信息客户端来显示已有的信息，合成新信息，或回复已有信息。

命名空间

QtMobility API 放在 QtMobility 的命名空间. 这是为以后Mobility APIs 集成到 Qt做准备. 参考Quickstart guide中的例子体会这个命名空间如何影响使用QtMobility的开发development.

简介

Messaging库提供了一系列API来获取信息数据。提供了查询检索信息，文件夹和账号的接口，也有创建，修改和存储信息的接口。提供了一个统一处理和储存短信（sms），彩信（MMS），邮件（MIME Email，TNEF Email）的接口。能处理 不完整提取的信息。

Messaging库提供了接口来发送信息，提取信息包含的数据，合并,回复,转发信息，调用本地应用来显示信息。

当存储的信息，增加，移除和更新的时候可以获得通知。

Messaging库提供了信息数据的统一访问，支持多个应用的并行访问，并不依赖于具体设备上数据的存储方式。

在 Microsoft Windows上，需要安装MAPI头文件，建议通过安装Visual Studio Express来做这件事。

在Linux上，需要安装QMF，同时设置两个环境变量，QMF_INCLUDEDIR设为messagingframework/src/libraries/qtopiamail ， QMF_LIBDIR设为build/messagingframework/image/lib。

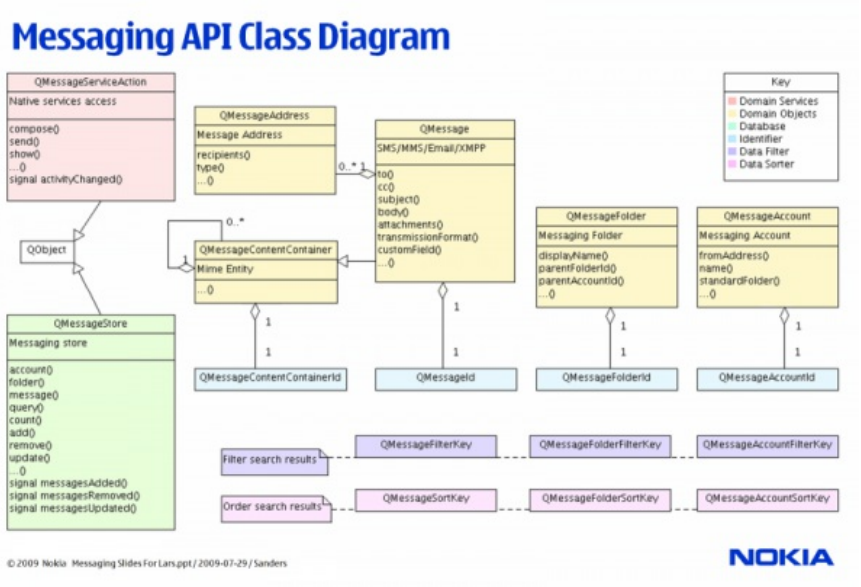

图1: Messaging 类图

(C) Copyright Nokia 2013. All rights reserved.

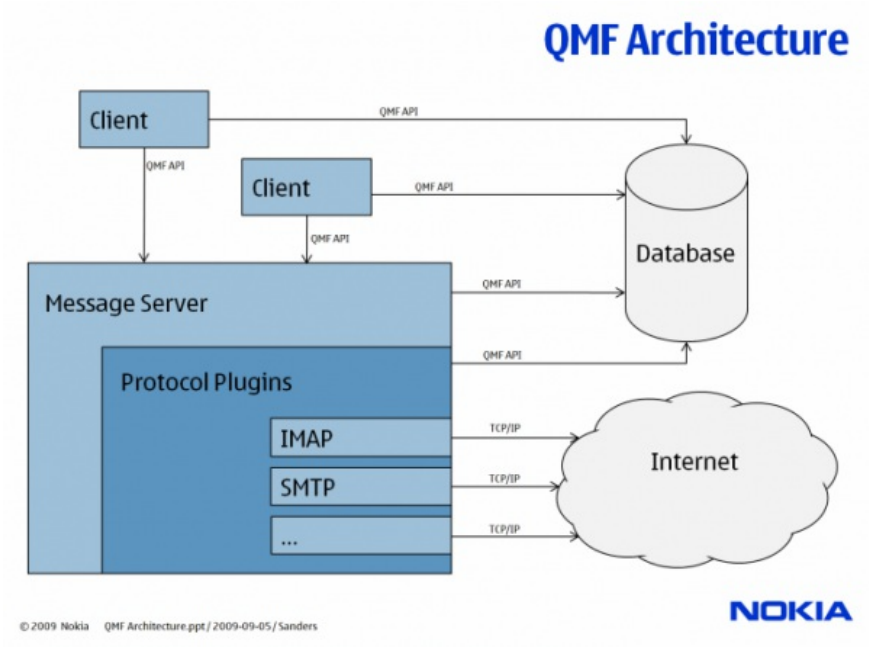

图2: Qt Messaging框架 架构

参考文档

主要的类

QMessage	提供信息相关服务的接口类
QMessageAccount	表示一个账号
QMessageAccountFilter	定义检索参数， 用来检索符合条件的账号
QMessageAccountId	QMessageAccount 的唯一标识UID
QMessageAccountSortOrder	定义排序的参数， 用来排列账号检索结果
QMessageAddress	表示信息地址的接口
QMessageContentContainer	表示信息内容的接口，支持互联网格式（MIME）和传送中性封装格式（TNEF）的内容，并支持不完整的内容。
QMessageContentContainerId	互联网格式的QMessageContentContainer条目的唯一标识
QMessageDataComparator	包含一些类型，使用这些类型的自定义值来匹配信息。
QMessageFilter	定义查找参数，用来查找信息。
QMessageFolder	表示信息文件夹的接口
QMessageFolderFilter	定义查询参数，用来查找文件夹
QMessageFolderId	信息文件夹的唯一标识UID
QMessageFolderSortOrder	定义排序的参数，用来排列文件夹检索结果
QMessageId	信息的的唯一标识UID
QMessageManager	存储,检索信息,文件夹和账号的主要接口
QMessageService	调用信息服务操作的接口
QMessageSortOrder	定义排序的参数，用来排列信息检索结果

组合和操作信息

Messaging库提供了一系列类来组合和操作信息数据。所有信息类型表示成统一的格式。信息可以由多个部分组成。 用以组合和操作的信息的类如下：

QMessageId	信息的UID
QMessage	一个设计的当的操作信息的接口
QMessageContentContainer	处理互联网媒质（MIME）的接口
QMessageAddress	操作信息地址字符串的接口

信息账号

Messaging库提供访问账号信息的类，接收和发送都可以处理。

信息账号相关的类如下：

QMessageAccountId	账号的UID
QMessageAccount	保存账号的属性
QMessageFolderId	文件夹的UID
QMessageFolder	表示一个文件夹中的信息集合

排序和过滤功能

Messaging库提供了一系列排序和过滤方法来辅助检索信息。这避免了依赖其他的库来管理信息数据。也有利于很高的利用底层的存储机制。 过滤操作通过设定过滤条件来实现，指定某个需要匹配的值或属性，设定需要检索的值与设定值之间的关系。 过滤器可以通过逻辑组合来生成复杂的过滤器。 Messaging库中排序和过滤相关的类如下：

QMessageManager	存储,检索信息,文件夹和账号的主要接口
QMessageFilter	定义查询参数，用来查找信息
QMessageSortOrder	定义排序的参数，用来排列信息检索结果
QMessageFolderFilter	定义查询参数，用来查找文件夹
QMessageFolderSortOrder	定义排序的参数，用来排列文件夹检索结果
QMessageAccountFilter	定义查询参数，用来查找账号
QMessageAccountSortOrder	定义排序的参数，用来排列账号检索结果

如何使用排序方法

下面是一个使用排序方法的例子，可以在发送信息的例子中找到。

(C) Copyright Nokia 2013. All rights reserved.

```
QMessageIdList ids = QMessageManager().queryMessages(QMessageFilter(),
    QMessageSortOrder::byReceptionTimeStamp(Qt::DescendingOrder), 100, 0);
```

如何使用过滤方法

下面的代码截取自Keep In Touch 例子。这个例子演示了如何使用过滤器达到想要的效果。你可以看到过滤器由逻辑运算连接，类似于C语言中的位操作。使用这种方法我们可以构造更复杂的检索条件。

```
QMessageFilter includeFilter(QMessageFilter::byTimeStamp(minimumDate,
        QMessageDataComparator::GreaterThanEqual));
QMessageFilter excludeFilter(QMessageFilter::byTimeStamp(maximumDate,
        QMessageDataComparator::GreaterThanEqual));
QMessageFilter outgoingFilter(QMessageFilter::byStatus(QMessage::Incoming,
        QMessageDataComparator::Excludes));

// Search for messages containing addresses to exclude
service.queryMessages(outgoingFilter & excludeFilter);

// Create the filter needed to locate messages to search for addresses
inclusionFilter = (outgoingFilter & includeFilter & ~excludeFilter);
```

信息服务

Messaging库提供了QMessageService 类，使用这个类来获得系统本地的信息服务，通过本地的应用实现 发送信息，提取信息中的数据，构造和显示信息等功能。

服务操作的状态通过发送信号（signal）来传递。

QMessageService类之外的方法不应调用网络通信，就是说只有QMessageService 的方法负责初始化网络连接 和随后的信息数据访问。（注意：对于windows桌面应用和有些移动平台并不是这样，有时客户端不能控制网络的激活。）

Messaging库访问信息数据的并行机制，并保证信息数据不会在多个客户端并行访问时损坏， 但是不保证并行访问的效率。

单个客户端可以通过多线程访问信息数据，但是同时只有一个线程能访问到数据。

抽象的信息存储机制

Messaging库将存储信息数据的方法抽象出来了。

QML Messaging Elements

- MessageModel
- MessageFilter
- MessageUnionFilter
- MessageIntersectionFilter

例子

Keep In Touch

Keep In Touch 这个例子演示了如何从系统中检索有用信息。

Service Actions

Service Actions这个例子演示如何使用 Qt Mobility Messaging API来构造信息，发送信息，显示，查找和检索信息，以及对信息事件反馈。

相关链接

- Qt Mobility 开发
- Qt 开发
- Qt Quick (Chinese)

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.