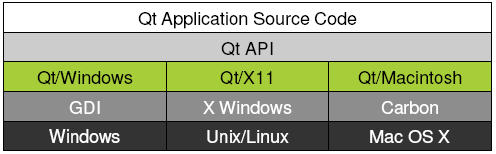
<http://blog.csdn.net/spwper/article/details/51316191>

版权声明：本文为博主原创文章，未经博主允许不得转载。

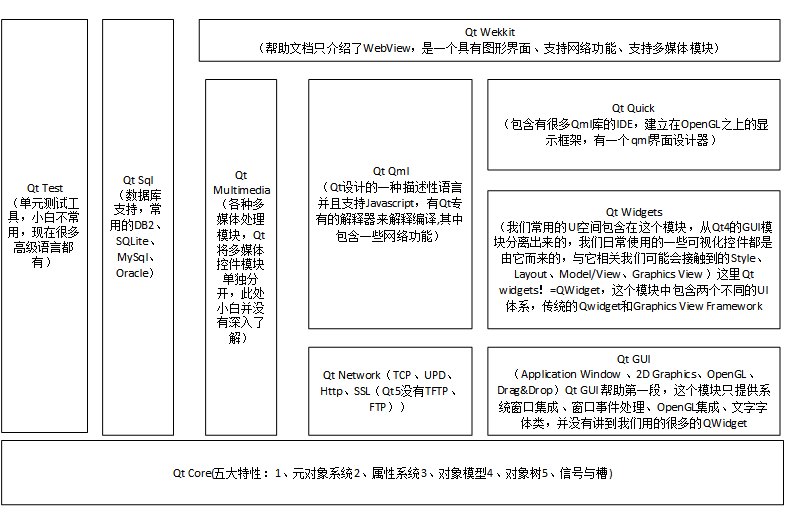
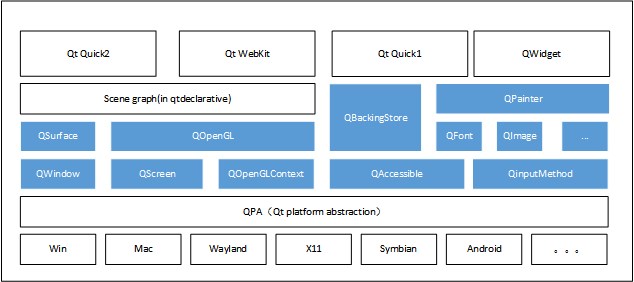
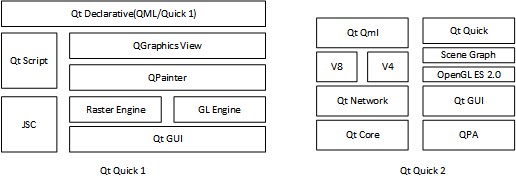
小白自工作就接触Qt，一直都在使用Qt5.3.1版本，所以没有经历过大牛们把项目从Qt4程序到Qt5的烦恼，没准以后会碰到。对Qt所有的丰富的API表示惊叹，对于Qt的框架及模块认识也是极为模糊的，文中有不对之处希望大牛们打脸（虽然脸都已经被打肿了。。。。）   
1、Qt的跨平台性（Linux、Unix、、Mac OS、Android、IOS）   
**简而言之：Qt的功能是建立在它所支持的平台底层API之上的。**   


* Qt/Windows对应的是GDI（Graphics Device Interface），很容易就知道它是Qt封装的与GDI交流的API。而GDI是Windows下图形绘制及显示的主要API。
* Qt/X11同样是对X Windows的编程接口。Linux本身是没有操作界面的，目前主流带有操作界面的Linux系统都是基于X WIndows的，已经列入了Linux发行标准软件了。X Windows是C/S框架的（就是所有的鼠标事件传递到X Windows的服务端处理之后对应在屏幕上显示内容），Qt/X11使用Xlib直接与X服务器通信的。（GDI是集成到Windows内核的，而X Windows只是Linux系统的应用软件而已）
* Qt/Macintosh就不用说了。Carbon是Mac OS X下应用程序开发环境，包含了应用程序的图形部分。目前主流的Cocoa好像使用更好。   
  上图主要说明的是UI部分，Qt跨平台都是调用本地的图形风格。

三种操作系统都支持主流的文件系统（文件系统并不是操作系统固有的，文件系统描述的硬盘里面文件存储和查找的方式），通信都是采用socket框架（socket是网络时代的产物，那时候的操作系统已经很成熟了）。至此可以大致了解了Qt跨平台的基本原理了。

## Qt框架

在Qt帮助里面搜索All modules可以清楚的看到Qt的模块分布。

1. Qt Essentials（Qt的基本模块）这里指的是开发者会经常使用到的稳定模块。
2. 2.Qt Add-Ons (扩展模块)扩展模块中针对移动端做的模块比较多，常用的Qt Bluetooth、Qt Positioning、Qt Sensors。UI方面的常用的有Qt Graphical Effects、Qt Image Formats、Qt SVG。新增的PC硬件资源的好用的Qt Print Support、Qt Serial Port，原来在多媒体模块有对Camera、Screen、Audio Device 的支持。Qt XML小白用的表较少，貌似在数据处理方面很好用。Qt Quick Widgets小白非常喜欢，因为小白去年写的UI基本是基于QWidget的，想扩展Qml界面这个非常方便。对于想在Qt上开发3D的Qt Canva 3D（5.5才有的），这个相对使用OPENGL入门更简单一些。
3. 还有一些Value-Add、Technology Preview 、 Tool modules都不太影响框架的认识。   
   **基本模块结构图**   
      
   小白这里只列举了一下常用到的模块结构分部，详细的还是需要参考帮助文档。   
   在Qt Widgets模块中，这里说明一下QGraphics Scene与QWidget是Qt两个不同的UI体系（前者基于OpenGL后者是基于原始的绘图设机制），发展源头是不一样的，详细可以看下这个博客[绘图设备和Graphics View Framework](http://blog.csdn.net/seanyxie/article/details/5825064)   
   **图形界面库的架构**   
      
   这个图里面包含Qt GUI以及Qt Widget的部分组成，其中蓝色块表示包含在GUI模块中的，可以参考一下上图中的Qt GUI模块的功能描述，并不包含所有的显示部分，想要仔细了解可以查阅帮助文档。   
   从这里也可以更加直观的看到两个不同的UI体系是怎么架构起来的。需要注意的是，Qt Quick1是从QPainter体系设计出来的。下面我们会对Qt Quick1与Qt Quick2的架构说明。   
   **Qt Qml和Qt Quick 框架**   
   QML：（Qt Meta-Object Language）强调指出这是一种语言，描述应用的声明式语言，什么语法规则什么啊，目前有很多脚本语言都是描述性的。   
   Qt Qml：是QML\*\*应用开发以及标准库的框架\*\*，并且提供了一个接口，允许开发者以自定义类型和继承Javascrip与C++代码方式来扩展QML语言。小白的理解就是，我们开发用的QML、Javascript、C++、Html混合编程时所需要的模块是在这里提供的。它是基于Qt专用引擎开发的（QML解释器）。   
   Qt Quick：Qml模块本书并不涉及显示，所有的图形处理都是有Quick模块来完成。经过了Qt GUI、OpenGL、Scene Graph三层封装。强调的是如何高效的绘制图形并按要求创建各种效果。所以里面包含了可视化类型（QML）、交互类型、动画、模型视图、粒子特效、渲染特效等。   
   分清楚这三个概念，来看看Quick1和Quick2的区别。   
      
   这里就可以很清楚为什么Quick1只能做一些变形旋转而没有更牛的特效了，因为QPainter确实不能做那个，大牛当然可以直接在Qt4下直接使用QOpenGL做特效。

Qt的基本框架这里大致还只是做了概念上的介绍，可以在阅读过程中查看帮助文档，边看变了解具体模块及模块中涉及到的类基本介绍。这里主要还是参考的《Qt5 编程入门》。以后还会不断的修补修补的。。。