

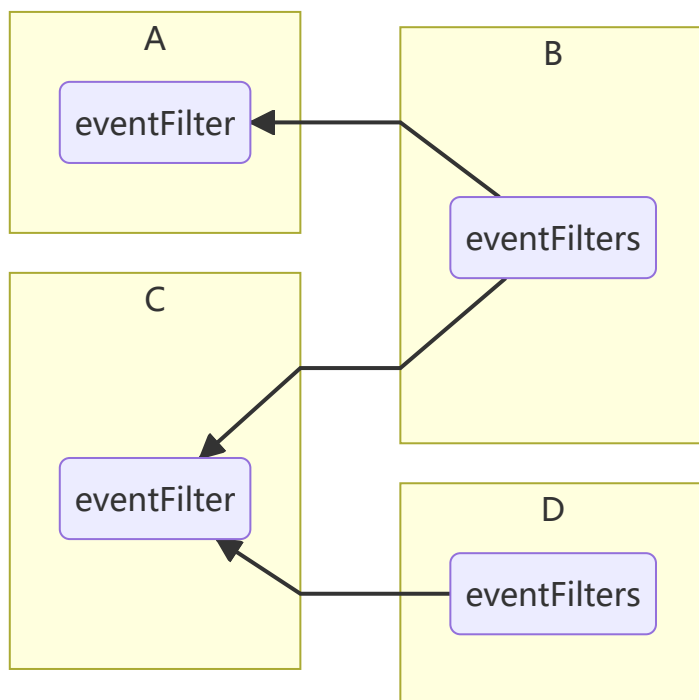
16 事件的处理

16.1 事件的调度

Qt的事件调度包括同步和异步两种模式。基于事件循环的事件调度就属于异步模式。应用程序一旦调用QApplication类的exec方法，即进入事件循环。事件循环先从Qt事件队列中弹出事件，若其为空，再从系统事件队列中弹出事件，并将其转换为Qt事件，压入Qt事件队列。另一种情形是直接调用QApplication类的sendEvent方法发送事件，所发送事件会被立即处理，这就是所谓的同步调度模式。QApplication类的sendEvent方法会调用QApplication类的notify方法完成对事件的派发。

16.2 事件的过滤

Qt通过所谓事件过滤器，让一个对象能够侦听并拦截递送给另一个对象的事件。事件过滤器的工作原理如下图所示：



A、B、C、D是四个QObject类或其子类的对象。在每个这样的对象内部，都有一个QObjectList类型的成员变量eventFilters，用于存放指向事件过滤器对象的指针。A对象在B对象上安装了事件过滤器，B对象eventFilters成员中即包含指向A对象的指针。类似地，C对象也可以在B对象上安装事件过滤器，因此B对象eventFilters成员中也包含指向C对象的指针。同时，C对象还在D对象上安装事件过滤器，D对象的eventFilters成员中同样包含了指向C对象的指针。经过如此一番操作，A对象和C对象是B对象的事件过滤器，而C对象同时还是D对象的事件过滤器。由此可见，一个对象可以同时拥有多个事件过滤器，而一个事件过滤器也可以同时被多个对象所拥有。

当一个事件被递送到B对象时，B对象会首先遍历其eventFilters成员中的事件过滤器指针，以过滤器被安装的反向顺序，依次调用每个过滤器对象的eventFilter方法，之后再将该事件交给自己的事件处理函数。事件过滤器对象的eventFilter方法在完成对事件的处理后，会返回一个bool类型的返回值。true表示该事件已被处理完毕，无需后续处理，Qt将直接进入对下一个事件的处理；false表示该事件还需继续处理，Qt会将该事件交给后续事件过滤器或目标对象进行处理。

16.3 事件的派发

QApplication类exec方法中的事件循环，一旦从事件队列中弹出了一个事件，即调用该类的notify方法完成对事件的派发。该方法会首先检查QApplication对象上是否被安装了事件过滤器，安装了则先调用事件过滤器，然后会滤除或合并一些事件，比如滤除失效组件上的鼠标事件，或合并相同区域上的绘制事件，最后将事件交给目标对象的event方法进行处理。

目标对象的event方法会首先检查该对象是否被安装了事件过滤器，安装了则先调用事件过滤器，然后根据事件的具体类型，调用与之对应的事件处理函数，比如mousePressEvent、focusOutEvent、resizeEvent、paintEvent、resizeEvent，等等。基类中的这些函数都被声明为虚函数。应用程序的设计者可以在自定义的子类中覆盖这些虚函数，用自己编写的代码实现对这些特定事件的处理逻辑。此外，还有一些不太常用的事件，Qt没有提供专门的事件处理函数。对于这种情况，我们可以直接覆盖基类的event方法，或借助事件过滤器予以解决。

16.4 事件的转发

事件目标组件的event方法在完成对事件的处理后，会返回一个bool类型的返回值。true表示该事件已被处理完毕，无需后续处理，Qt将直接进入对下一个事件的处理；false表示该事件还需继续处理，Qt会将该事件交给目标组件的父组件处理。另外，也可以通过事件对象本身的ignore或accept方法，将该事件标记为忽略或接受。被标记为忽略的事件将跳过与具体事件类型对应的事件处理函数，直接转发给目标组件的父组件进行处理。当然这仅限于在event方法中对诸如鼠标、滚轮、按键等特定事件的处理。

16.5 事件的处理

Qt提供了五种不同级别的事件处理方法。第一种方法是覆盖组件基类的mousePressEvent、focusOutEvent、resizeEvent、paintEvent、resizeEvent等虚函数。这是最常用、最简单的事件处理方式。第二种方法是在QApplication类的子类中覆盖其基类的notify方法，在事件被派发到目标对象前即完成处理。这种方法的功能十分强大，可以从源头上获得对事件的完全控制，但同一时间只能有一个QApplication类的子类对象被激活。第三种方法是为QApplication对象安装事件过滤器，从事件队列中弹出的所有事件都会先经过事件过滤器的eventFilter方法。这种方法的功能和覆盖QApplication类的notify方法一样强大，但无需继承QApplication类，而且可以给QApplication对象安装任意数量的事件过滤器。第四种方法是覆盖组件基类的event方法，对于不需要处理或不知道如何处理的事件，可以简单地将其交给基类的event方法，并返回其返回值。第五种方法是为特定的组件安装事件过滤器，在不扩展组件类型的前提下，定制其事件处理方式。