Atitit.图片木马 与 远程接口 监控 常用的api 标准化v2 q216

[1. 木马与远程接口 监控的常用的api 2](#_Toc10475)

[1.1. 文件复制 2](#_Toc1407)

[1.2. 屏幕定时截图 2](#_Toc16165)

[1.3. 邮件发送 2](#_Toc18622)

[1.4. 键盘监听 2](#_Toc23105)

[1.5. 远程上传代码与执行 2](#_Toc25858)

[1.6. 注册系统服务 2](#_Toc26705)

[1.7. 远程shell console 2](#_Toc3932)

[1.8. 、修改注册表 2](#_Toc29767)

[1.9. Eval功能 2](#_Toc12432)

[1.10. 控制鼠标键盘gui接口功能 2](#_Toc3846)

[1.11. 文档 图片文件读写监控 2](#_Toc31841)

[1.11.1. JDK1.6及之前版本: 基于Timer实现 2](#_Toc4617)

[1.11.2. JDK 1.7 及之后版本：基于WatchService实现 3](#_Toc4119)

[2. 参考 4](#_Toc4169)

# 木马与远程接口 监控的常用的api

## 文件复制

## 屏幕定时截图

## 邮件发送

## 键盘监听

## 远程上传代码与执行

## 注册系统服务

## 远程shell console

## 、修改注册表

## Eval功能

## 控制鼠标键盘gui接口功能

## 文档 图片文件读写监控

### ****JDK1.6及之前版本: 基于Timer实现****

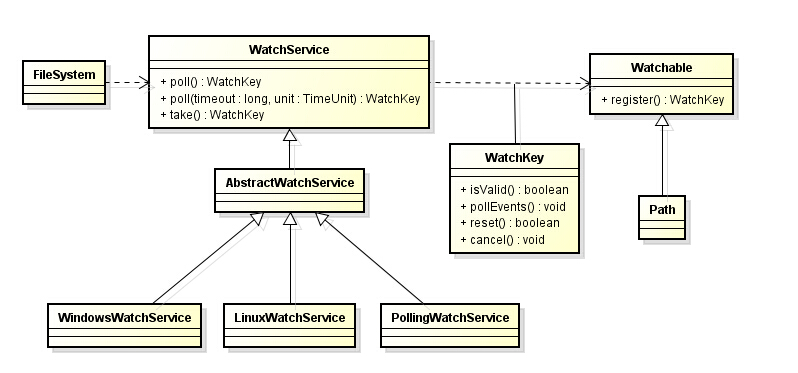
通过实现FileChangeObserver接口，该FileMonitor允许对任意文件添加任意多个Observer

通过比对时间戳来判断文件是否修改，若发生改动，则通知其Observer进行相应处理。

常用doc pic文件内容扫描

### ****JDK 1.7 及之后版本：基于WatchService实现****

Watch Service API可用于将指定目录注册到监视服务上。注册时须指定事件类型，如文件创建、修改、删除等。相关类图如下：



WatchService是监视服务接口，在不同系统上有不同的实现类。实现了Watchable接口的对象方可注册监视服务，java.nio.file.Path实现了此接口。WatchKey表示Watchable对象和WatchService的关联关系，在注册时被创建。注册完成后，WatchKey将被置为'ready'状态，直到下列三种情况之一发生：

1. WatchKey.cancel()被调用

2. 被监控的目录不存在或不可访问

3. WatchService对象被关闭

当文件改动发生时，WatchKey的状态将会被置为"signaled"然后被放入待处理队列中。WatchService提供了****三种从队列中获取WatchKeys的方式****：

1. poll - 返回队列中的一个key。如果没有可用的key，将立即返回null。

2. poll(long, TimeUnit) - 如果队列中存在可用的key则将之返回，否则在参数预置的时间内等待可用的key。TimeUnit用来指定前一个参数表示的时间是纳秒、毫秒或是其他的时间单位。

例子：final WatchKey watchKey = watchService.poll(1, TimeUnit.MINUTES);将会等待1分钟

3. take - 方法将会等待直到可用的key被返回。

****获取WatchKey后进行处理：****

1. 通过WatchKey.pollEvents()函数得到WatchEvents列表。

2. 对于每一个WatchEvent，可以通过kind()函数获得其改动类型。

3. 通过WatchEvent.context()函数得到发生该事件的文件名

4. 当该key的所有事件处理完成后，需要调用WatchKey.reset()方法把该key重置为ready状态。若不重置，该key将无法接收后续的改动。若reset返回false，表示该WatchKey不再合法，主循环可以退出。

总结，使用WatchService步骤如下：

1. 创建WatchService
2. 得到待检测目录的Path
3. 将目录登记到变化监测名单中
4. 执行WatchService的take()方法，直到WatchKey到来。
5. 得到WatchKey后遍历WatchEvent进行检测
6. 重置key准备下一个事件，继续等待

大多数文件系统实现包含了文件更改通知的本地支持，Watch Service API正是利用了文件系统的这种机制。若文件系统并不支持变更通知机制，Watch Service仍然会轮询文件系统，等待事件产生。

### C# asp 。net的的文件更改监视

。。。Like java ..use os support api

# 参考

****Java文件变更监控的两种实现 - Stay hungry, stay foolish. - ITeye技术网站.htm****