



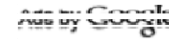
Unrated

Berkley Recruitment

IT Java Positions Available Contract & Permanent Roles.

Java Installer Builder

Easy to use, amazingly powerful Creates beautiful installers



Detect file modification

For a single file, a thread is launched to check the lastModified value and compare it with the previous value.

```
import java.util.*;
import java.io.*;

public abstract class FileWatcher extends TimerTask {
    private long timeStamp;
    private File file;

    public FileWatcher( File file ) {
        this.file = file;
        this.timeStamp = file.lastModified();
    }

    public final void run() {
        long timeStamp = file.lastModified();

        if( this.timeStamp != timeStamp ) {
            this.timeStamp = timeStamp;
            onChange(file);
        }
    }

    protected abstract void onChange( File file );
}

import java.util.*;
import java.io.*;

public class FileWatcherTest {
    public static void main(String args[]) {
        // monitor a single file
        TimerTask task = new FileWatcher( new File("c:/temp/text.txt") ) {
            protected void onChange( File file ) {
                // here we code the action on a change
                System.out.println( "File " + file.getName() + " have change !" );
            }
        };

        Timer timer = new Timer();
        // repeat the check every second
        timer.schedule( task , new Date(), 1000 );
    }
}
```

For a directory, a thread is launched where we keep the Files in a Map, we check the current lastModified value of a given file and compare it with the value stored in the Map. Also a special check is made to detect if a File is deleted.

```
import java.util.*;
import java.io.*;

public abstract class DirWatcher extends TimerTask {
    private String path;
    private File filesArray [];
    private HashMap dir = new HashMap();
    private DirFilterWatcher dfw;

    public DirWatcher(String path) {
        this(path, "");
    }

    public DirWatcher(String path, String filter) {
        this.path = path;
        dfw = new DirFilterWatcher(filter);
        filesArray = new File(path).listFiles(dfw);

        // transfer to the hashmap be used a reference and keep the
        // lastModified value
        for(int i = 0; i < filesArray.length; i++) {
            dir.put(filesArray[i], new Long(filesArray[i].lastModified()));
        }
    }

    public final void run() {
        HashSet checkedFiles = new HashSet();
        filesArray = new File(path).listFiles(dfw);

        // scan the files and check for modification/addition
        for(int i = 0; i < filesArray.length; i++) {
            Long current = (Long)dir.get(filesArray[i]);
            checkedFiles.add(filesArray[i]);
            if (current == null) {
                // new file
                dir.put(filesArray[i], new Long(filesArray[i].lastModified()));
                onChange(filesArray[i], "add");
            }
            else if (current.longValue() != filesArray[i].lastModified()){
                // modified file
                dir.put(filesArray[i], new Long(filesArray[i].lastModified()));
                onChange(filesArray[i], "modify");
            }
        }

        // now check for deleted files
        Set ref = ((HashMap)dir.clone()).keySet();
        ref.removeAll((Set)checkedFiles);
        Iterator it = ref.iterator();
        while (it.hasNext()) {
            File deletedFile = (File)it.next();
            dir.remove(deletedFile);
            onChange(deletedFile, "delete");
        }
    }

    protected abstract void onChange( File file, String action );
}
```

```
import java.io.*;

public class DirFilterWatcher implements FileFilter {
    private String filter;

    public DirFilterWatcher() {
        this.filter = "";
    }

    public DirFilterWatcher(String filter) {
        this.filter = filter;
    }

    public boolean accept(File file) {
        if ("".equals(filter)) {
            return true;
        }
        return (file.getName().endsWith(filter));
    }
}
```

The example watches the c:/temp folder for any activities on any *.txt files.

```
import java.util.*;
import java.io.*;

public class DirWatcherTest {
    public static void main(String args[]) {
        TimerTask task = new DirWatcher("c:/temp", "txt" ) {
            protected void onChange( File file, String action ) {
                // here we code the action on a change
                System.out.println
                    ( "File "+ file.getName() +" action: " + action );
            }
        };

        Timer timer = new Timer();
        timer.schedule( task , new Date(), 1000 );
    }
}
```

See also this [related howto](#) and this [one too](#).



If you find this article useful, consider making a small donation to show your support for this Web site and its content.



Written and compiled by Réal Gagnon ©1998-2009

[\[home \]](#)