获取系统分隔符的方法

1. 路径分隔符：

private static final String PATHSEPARATOR = System.getProperty("path.separator");

private static final String PATHSEPARATOR2 = File.pathSeparator;

1. 文件名称分隔符：

private static final String FILESEPARATOR = System.getProperty("file.separator");

private static final String FILESEPARATOR2 = File.separator;

1. 注意：java系统有一个配置文件，通过System类的静态方法getProperties方法即可获取。

对系统的配置文件进行遍历：

Properties properties = System.*getProperties*();

Set<Object> keySet = properties.keySet();

Iterator<Object> iterator = keySet.iterator();

**while** (iterator.hasNext()) {

String key = (String) iterator.next();

System.***out***.println(key.toString() + ":" + System.*getProperty*(key));

}

**查询好具体的key，可以通过System的静态方法getProperty方法获取对应的符号：**

System.out.println("Java的运行环境版本："+props.getProperty("java.version"));  
   System.out.println("Java的运行环境供应商："+props.getProperty("java.vendor"));  
   System.out.println("Java供应商的URL："+props.getProperty("java.vendor.url"));  
   System.out.println("Java的安装路径："+props.getProperty("java.home"));  
   System.out.println("Java的虚拟机规范版本："+props.getProperty("java.vm.specification.version"));  
   System.out.println("Java的虚拟机规范供应商："+props.getProperty("java.vm.specification.vendor"));  
   System.out.println("Java的虚拟机规范名称："+props.getProperty("java.vm.specification.name"));  
   System.out.println("Java的虚拟机实现版本："+props.getProperty("java.vm.version"));  
   System.out.println("Java的虚拟机实现供应商："+props.getProperty("java.vm.vendor"));  
   System.out.println("Java的虚拟机实现名称："+props.getProperty("java.vm.name"));  
   System.out.println("Java运行时环境规范版本："+props.getProperty("java.specification.version"));  
   System.out.println("Java运行时环境规范供应商："+props.getProperty("java.specification.vender"));  
   System.out.println("Java运行时环境规范名称："+props.getProperty("java.specification.name"));  
   System.out.println("Java的类格式版本号："+props.getProperty("java.class.version"));  
   System.out.println("Java的类路径："+props.getProperty("java.class.path"));  
   System.out.println("加载库时搜索的路径列表："+props.getProperty("java.library.path"));  
   System.out.println("默认的临时文件路径："+props.getProperty("java.io.tmpdir"));  
   System.out.println("一个或多个扩展目录的路径："+props.getProperty("java.ext.dirs"));  
   System.out.println("操作系统的名称："+props.getProperty("os.name"));  
   System.out.println("操作系统的构架："+props.getProperty("os.arch"));  
   System.out.println("操作系统的版本："+props.getProperty("os.version"));  
   System.out.println("文件分隔符："+props.getProperty("file.separator"));   //在 unix 系统中是＂／＂  
   System.out.println("路径分隔符："+props.getProperty("path.separator"));   //在 unix 系统中是＂:＂  
   System.out.println("行分隔符："+props.getProperty("line.separator"));   //在 unix 系统中是＂/n＂  
   System.out.println("用户的账户名称："+props.getProperty("user.name"));  
   System.out.println("用户的主目录："+props.getProperty("user.home"));  
   System.out.println("用户的当前工作目录："+props.getProperty("user.dir"));

<http://www.blogjava.net/void241/archive/2009/06/04/279971.html>

1. （1）String path1 = "E:\\test\\aaa.txt";

**之所以用两个\，因为第一个\是用来转义。**

（2）String path2 = "E:"+ File.separator+"test"+File.separator+"aaa.txt";

可以跨平台，动态获取相应系统下的分隔符。

（3） String path3 = "E:/test/aaa.txt";

5.