JAVA.LANG.SECURITYMANAGER.CHECKDELETE METHOD

Description

The **java.lang.SecurityManager.checkDelete*Stringfile*** method throws a SecurityException if the calling thread is not allowed to delete the specified file. This method is invoked for the current security manager by the delete method of class File.

This method calls checkPermission with the FilePermission*file, " delete "* permission. If you override this method, then you should make a call to super.checkDelete at the point the overridden method would normally throw an exception.

Declaration

Following is the declaration for **java.lang.SecurityManager.checkDelete** method

```
public void checkDelete(String file)
```

Parameters

- **file** -- the system-dependent filename.

Return Value

This method does not return a value.

Exception

- **SecurityException** -- if the calling thread does not have permission to delete the file.
- **NullPointerException** -- if the file argument is null

Example

Our examples require that the permissions for each command is blocked. A new policy file was set that allows only the creating and setting of our Security Manager. The file is in C:/java.policy and contains the following text:

```
grant {
  permission java.lang.RuntimePermission "setSecurityManager";
  permission java.lang.RuntimePermission "createSecurityManager";
  permission java.lang.RuntimePermission "usePolicy";
};
```

The following example shows the usage of lang.SecurityManager.checkDelete method.

```
package com.tutorialspoint;

public class SecurityManagerDemo {

   public static void main(String[] args) {

   // set the policy file as the system securuty policy
   System.setProperty("java.security.policy", "file:/C:/java.policy");

   // create a security manager
   SecurityManager sm = new SecurityManager();

   // set the system security manager
   System.setSecurityManager(sm);

   // perform the check
```

```
    sm.checkDelete("test.txt");

    // print a message if we passed the check
    System.out.println("Allowed!");
    }
}
```

Let us compile and run the above program, this will produce the following result:

```
Exception in thread "main" java.security.AccessControlException: access denied
(java.io.FilePermission test.txt delete)
```

Loading [MathJax]/jax/output/HTML-CSS/jax.js