

# SECURITYMANAGER CHECKPERMISSION METHOD

[http://www.tutorialspoint.com/java/lang/securitymanager\\_checkpermission.htm](http://www.tutorialspoint.com/java/lang/securitymanager_checkpermission.htm)

Copyright © tutorialspoint.com

## Description

The **java.lang.SecurityManager.checkPermission***Permissionperm* method throws a **SecurityException** if the requested access, specified by the given permission, is not permitted based on the security policy currently in effect. This method calls **AccessController.checkPermission** with the given permission.

## Declaration

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

```
public void checkPermission(Permission perm)
```

## Parameters

- **perm** -- the requested permission.

## Return Value

This method does not return a value.

## Exception

- **SecurityException** -- if access is not permitted based on the current security policy.
- **NullPointerException** -- if the permission 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.checkPermission** method.

```
package com.tutorialspoint;  
  
import java.io.FilePermission;  
  
public class SecurityManagerDemo extends SecurityManager {  
  
    public static void main(String[] args) {  
  
        // set the policy file as the system security policy  
        System.setProperty("java.security.policy", "file:/C:/java.policy");  
  
        // create a security manager  
        SecurityManagerDemo sm = new SecurityManagerDemo();  
  
        // set the system security manager  
        System.setSecurityManager(sm);  
  
        // perform the check  
        sm.checkPermission(new FilePermission("test.txt", "read,write"));  
    }  
}
```

```
// 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 read,write)
```

```
Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js
```