

# SECURITYMANAGER CHECKPROPERTIESACCESS METHOD

[http://www.tutorialspoint.com/java/lang/securitymanager\\_checkpropertiesaccess.htm](http://www.tutorialspoint.com/java/lang/securitymanager_checkpropertiesaccess.htm) Copyright © tutorialspoint.com

## Description

The **java.lang.SecurityManager.checkPropertiesAccess** method throws a `SecurityException` if the calling thread is not allowed to access or modify the system properties. This method is used by the `getProperties` and `setProperties` methods of class `System`.

This method calls `checkPermission` with the `PropertyPermission` " \* ", " read, write " permission. If you override this method, then you should make a call to `super.checkPropertiesAccess` at the point the overridden method would normally throw an exception.

## Declaration

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

```
public void checkPropertiesAccess()
```

## Parameters

- NA

## Return Value

This method does not return a value.

## Exception

- **SecurityException** -- if the calling thread does not have permission to access or modify the system properties.

## 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.checkPropertiesAccess` method.

```
package com.tutorialspoint;  
  
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.checkPropertiesAccess();

// 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.util.PropertyPermission * read,write)
```

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