

SECURITYMANAGER CHECKPROPERTYACCESS METHOD

http://www.tutorialspoint.com/java/lang/securitymanager_checkpropertyaccess.htm Copyright © tutorialspoint.com

Description

The **java.lang.SecurityManager.checkPropertyAccessStringkey** method throws a **SecurityException** if the calling thread is not allowed to access the system property with the specified key name. This method is used by the **getProperty** method of class **System**.

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

Declaration

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

```
public void checkPropertyAccess(String key)
```

Parameters

- **key** -- a system property key.

Return Value

This method does not return a value.

Exception

- **SecurityException** -- if the calling thread does not have permission to access the specified system property.
- **NullPointerException** -- if the key argument is null.
- **IllegalArgumentException** -- if key is empty.

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.checkPropertyAccess** 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.checkPropertyAccess("java.runtime.name");

// 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 java.runtime.name read)
Loading [MathJax]/jax/output/HTML-CSS/jax.js
```