

# SECURITYMANAGER CHECKTOPELLEVELWINDOW METHOD

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

## Description

The **java.lang.SecurityManager.checkTopLevelWindow** method returns false if the calling thread is not trusted to bring up the top-level window indicated by the window argument. In this case, the caller can still decide to show the window, but the window should include some sort of visual warning. If the method returns true, then the window can be shown without any special restrictions.

See class Window for more information on trusted and untrusted windows. This method calls checkPermission with the AWTPermission "showWindowWithoutWarningBanner" permission, and returns true if a SecurityException is not thrown, otherwise it returns false.

If you override this method, then you should make a call to super.checkTopLevelWindow at the point the overridden method would normally return false, and the value of super.checkTopLevelWindow should be returned.

## Declaration

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

```
public boolean checkTopLevelWindow(Object window)
```

## Parameters

- **window** -- the new window that is being created.

## Return Value

This method returns *true* if the calling thread is trusted to put up top-level windows; *false* otherwise.

## Exception

- **NullPointerException** -- if the window 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.checkTopLevelWindow 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);

// check the top level window
boolean checked = sm.checkTopLevelWindow("Window");
System.out.println("" + checked);

}
}
```

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

false

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