

# SECURITYMANAGER GETSECURITYCONTEXT METHOD

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

Copyright © tutorialspoint.com

## Description

The **java.lang.SecurityManager.getSecurityContext** method creates an object that encapsulates the current execution environment. The result of this method is used, for example, by the three-argument checkConnect method and by the two-argument checkRead method. These methods are needed because a trusted method may be called on to read a file or open a socket on behalf of another method. The trusted method needs to determine if the other *possiblyuntrusted* method would be allowed to perform the operation on its own. The default implementation of this method is to return an AccessControlContext object.

## Declaration

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

```
public Object getSecurityContext()
```

## Parameters

- NA

## Return Value

This method returns an implementation-dependent object that encapsulates sufficient information about the current execution environment to perform some security checks later.

## Exception

- NA

## 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.getSecurityContext method.

```
package com.tutorialspoint;  
  
public class SecurityManagerDemo {  
  
    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  
        SecurityManager sm = new SecurityManager();  
  
        // set the system security manager  
        System.setSecurityManager(sm);  
  
        // get the security context  
        Object con = sm.getSecurityContext();  
    }  
}
```

```
// print the class context
System.out.println("'" + con);

}
```

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

```
java.security.AccessControlContext@5f186fab
Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js
```