

**Throws:**

`java.lang.SecurityException` - if called by an `AccessControlContext` that has not been granted the “setPolicy” `SecurityPermission`.

`java.lang.UnsupportedOperationException` - if the state of the policy context whose interface is this `PolicyConfiguration` Object is “deleted” when this method is called. This exception is also thrown when completing the commit would cause there to be two or more `inService` and linked policy contexts with different principal-to-role mappings.

`PolicyContextException` - if the implementation throws a checked exception that has not been accounted for by the commit method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown `PolicyContextException`.

**delete()**

```
public void delete()
    throws PolicyContextException
```

Causes all policy statements to be deleted from this `PolicyConfiguration` and sets its internal state such that calling any method, other than `delete`, `getContextID`, or `inService` on the `PolicyConfiguration` will be rejected and cause an `UnsupportedOperationException` to be thrown.

This operation has no affect on any linked `PolicyConfigurations` other than removing any links involving the deleted `PolicyConfiguration`.

**Throws:**

`java.lang.SecurityException` - if called by an `AccessControlContext` that has not been granted the “setPolicy” `SecurityPermission`.

`PolicyContextException` - if the implementation throws a checked exception that has not been accounted for by the `delete` method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown `PolicyContextException`.

**getContextID()**

```
public java.lang.String getContextID()
    throws PolicyContextException
```

This method returns this object’s policy context identifier.

**Returns:** this object’s policy context identifier.

**Throws:**

`java.lang.SecurityException` - if called by an `AccessControlContext` that has not been granted the “setPolicy” `SecurityPermission`.

`PolicyContextException` - if the implementation throws a checked exception that has not been accounted for by the `getContextID` method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown `PolicyContextException`.

**inService()**

```
public boolean inService()
    throws PolicyContextException
```

This method is used to determine if the policy context whose interface is this `PolicyConfiguration` Object is in the “inService” state.

**Returns:** true if the state of the associated policy context is “inService”; false otherwise.