4.9 Missing Policy Contexts

A Policy provider must return that a tested permission has not been granted if it acquires a non-null policy context identifier by calling getContextID on the PolicyContext class and the inService method of the PolicyConfigurationFactory associated with the provider would return false if called with the policy context identifier.

4.10 Default Policy Context

The default policy context contains the policy statements that apply to the JRE independent of the policy contexts defined as the result of the deployment of modules or applications in containers. The policy context identifier of the default policy context is the null value. The default policy context is never linked to another PolicyConfiguration, and as such does not share the principal-to-role mapping of any other policy context.

A Policy provider must include the policy statements of the default policy context in every access determination it performs. A Policy provider that either does not call PolicyContext.getContextID, or does so and acquires the identifier of the default policy context, must use only the policy statements of the default policy context to perform its access determination.

4.11 Policy Compatibility Requirements

To be compatible with this contract, every JRE of a J2EE 1.4 application server must perform all of the policy decisions defined by this contract by interacting with the java.security.Policy instance available in the JRE via the java.security.Policy.getPolicy method. Every JRE of a J2EE 1.3 application server must perform all of the policy decisions defined by this contract by interacting with the javax.security.auth.Policy instance available in the JRE via the javax.security.auth.Policy.getPolicy method.

If an application server or JRE employs a custom SecurityManager, the necessary reliance on Policy object may be accomplished by ensuring that the custom SecurityManager relies on the appropriate (as defined above) Policy object for all of the policy decisions defined by this contract.