**Signed Applet**

Java applets are contained within a “sandbox” by default and due to security reason. A signed Applet is a trusted Applet. This signed Applet can perform any action within sandbox and outsides it boundaries. For example, reading, writing and deleting file on local drives or accessing different system parameters.  
  
Earlier versions of browsers have no support for signed Applet and do not allow operations outside sandbox. To sign an Applet, applet’s code must be digitally singed with digital ID. The unsigned Applet can request for privilege to access resources outside sand box.

Name:  62.JPG
Views: 52
Size:  20.1 KB

# What Applets Can and Cannot Do

Java applets are loaded on a client when the user visits a page containing an applet. The security model behind Java applets has been designed with the goal of protecting the user from malicious applets.

Applets that are not signed using a security certificate are considered to be untrusted and referred to as unsigned applets. When running on a client, unsigned applets operate within a security sandbox that allows only a set of safe operations.

Applets can be signed using a security certificate to indicate that they come from a trusted source. Signed applets operate outside the security sandbox and have extensive capabilities to access the client. A signed applet will run outside the security sandbox only if the user accepts the applet's security certificate. If the user refuses to accept the certificate, the applet will run within the security sandbox similar to an unsigned applet.

With recent improvements to the Java Plug-in software, unsigned applets launched using Java Network Launch Protocol (JNLP) can safely access the client with the user's permission. It is recommended that you launch your applet using JNLP to leverage expanded capabilities and improve user experience. See [Deploying an Applet](http://docs.oracle.com/javase/tutorial/deployment/applet/deployingApplet.html) for step by step instructions on applet deployment.

In this, topic we will discuss the security restrictions and capabilities of applets.

## Unsigned Applets

Unsigned applets can perform the following operations:

* They can make network connections to the host they came from.
* They can easily display HTML documents using the showDocument method of the java.applet.AppletContext class.
* They can invoke public methods of other applets on the same page.
* Applets that are loaded from the local file system (from a directory in the user's CLASSPATH) have none of the restrictions that applets loaded over the network do.
* They can read secure system properties. See [System Properties](http://docs.oracle.com/javase/tutorial/deployment/doingMoreWithRIA/properties.html) for a list of secure system properties.
* When launched by using JNLP, unsigned applets can also perform the following operations:
  + They can open, read, and save files on the client.
  + They can access the shared system-wide clipboard.
  + They can access printing functions.
  + They can store data on the client, decide how applets should be downloaded and cached, and much more. See [JNLP API](http://docs.oracle.com/javase/tutorial/deployment/doingMoreWithRIA/jnlpAPI.html) for more information about developing applets by using the JNLP API.

Unsigned applets cannot perform the following operations:

* They cannot access client resources such as the local filesystem, executable files, system clipboard, and printers.
* They cannot connect to or retrieve resources from any third party server (any server other than the server it originated from).
* They cannot load native libraries.
* They cannot change the SecurityManager.
* They cannot create a ClassLoader.
* They cannot read certain system properties. See [System Properties](http://docs.oracle.com/javase/tutorial/deployment/doingMoreWithRIA/properties.html) for a list of forbidden system properties.

## Signed Applets

Signed applets do not have the security restrictions that are imposed on unsigned applets and can run outside the security sandbox.

**Note:**  JavaScript code is treated like unsigned code. When a signed applet is accessed from JavaScript code in an HTML page, the applet is executed within the security sandbox. This implies that the signed applet essentially behaves likes an unsigned applet.