| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ClassLoader.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/lang/ClassFormatError.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/ClassNotFoundException.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/ClassLoader.html)    [**NO FRAMES**](http://docs.google.com/ClassLoader.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#17dp8vu) |

## **java.lang**

Class ClassLoader

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.lang.ClassLoader**

**Direct Known Subclasses:** [SecureClassLoader](http://docs.google.com/java/security/SecureClassLoader.html)

public abstract class **ClassLoader**extends [Object](http://docs.google.com/java/lang/Object.html)

A class loader is an object that is responsible for loading classes. The class ClassLoader is an abstract class. Given the [binary name](#_3znysh7) of a class, a class loader should attempt to locate or generate data that constitutes a definition for the class. A typical strategy is to transform the name into a file name and then read a "class file" of that name from a file system.

Every [Class](http://docs.google.com/java/lang/Class.html) object contains a [reference](http://docs.google.com/java/lang/Class.html#getClassLoader()) to the ClassLoader that defined it.

Class objects for array classes are not created by class loaders, but are created automatically as required by the Java runtime. The class loader for an array class, as returned by [Class.getClassLoader()](http://docs.google.com/java/lang/Class.html#getClassLoader()) is the same as the class loader for its element type; if the element type is a primitive type, then the array class has no class loader.

Applications implement subclasses of ClassLoader in order to extend the manner in which the Java virtual machine dynamically loads classes.

Class loaders may typically be used by security managers to indicate security domains.

The ClassLoader class uses a delegation model to search for classes and resources. Each instance of ClassLoader has an associated parent class loader. When requested to find a class or resource, a ClassLoader instance will delegate the search for the class or resource to its parent class loader before attempting to find the class or resource itself. The virtual machine's built-in class loader, called the "bootstrap class loader", does not itself have a parent but may serve as the parent of a ClassLoader instance.

Normally, the Java virtual machine loads classes from the local file system in a platform-dependent manner. For example, on UNIX systems, the virtual machine loads classes from the directory defined by the CLASSPATH environment variable.

However, some classes may not originate from a file; they may originate from other sources, such as the network, or they could be constructed by an application. The method [defineClass](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20byte%5B%5D,%20int,%20int)) converts an array of bytes into an instance of class Class. Instances of this newly defined class can be created using [Class.newInstance](http://docs.google.com/java/lang/Class.html#newInstance()).

The methods and constructors of objects created by a class loader may reference other classes. To determine the class(es) referred to, the Java virtual machine invokes the [loadClass](http://docs.google.com/java/lang/ClassLoader.html#loadClass(java.lang.String)) method of the class loader that originally created the class.

For example, an application could create a network class loader to download class files from a server. Sample code might look like:

ClassLoader loader = new NetworkClassLoader(host, port);  
 Object main = loader.loadClass("Main", true).newInstance();  
  . . .

The network class loader subclass must define the methods [findClass](http://docs.google.com/java/lang/ClassLoader.html#findClass(java.lang.String)) and loadClassData to load a class from the network. Once it has downloaded the bytes that make up the class, it should use the method [defineClass](http://docs.google.com/java/lang/ClassLoader.html#defineClass(byte%5B%5D,%20int,%20int)) to create a class instance. A sample implementation is:

class NetworkClassLoader extends ClassLoader {  
 String host;  
 int port;  
  
 public Class findClass(String name) {  
 byte[] b = loadClassData(name);  
 return defineClass(name, b, 0, b.length);  
 }  
  
 private byte[] loadClassData(String name) {  
 // load the class data from the connection  
  . . .  
 }  
 }

#### Binary names

Any class name provided as a [String](http://docs.google.com/java/lang/String.html) parameter to methods in ClassLoader must be a binary name as defined by the [Java Language Specification](http://java.sun.com/docs/books/jls/).

Examples of valid class names include:

"java.lang.String"  
 "javax.swing.JSpinner$DefaultEditor"  
 "java.security.KeyStore$Builder$FileBuilder$1"  
 "java.net.URLClassLoader$3$1"

**Since:** 1.0 **See Also:**[resolveClass(Class)](http://docs.google.com/java/lang/ClassLoader.html#resolveClass(java.lang.Class))

| **Constructor Summary** | |
| --- | --- |
| protected | [**ClassLoader**](http://docs.google.com/java/lang/ClassLoader.html#ClassLoader())()            Creates a new class loader using the ClassLoader returned by the method [getSystemClassLoader()](http://docs.google.com/java/lang/ClassLoader.html#getSystemClassLoader()) as the parent class loader. |
| protected | [**ClassLoader**](http://docs.google.com/java/lang/ClassLoader.html#ClassLoader(java.lang.ClassLoader))([ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) parent)            Creates a new class loader using the specified parent class loader for delegation. |

| **Method Summary** | |
| --- | --- |
| void | [**clearAssertionStatus**](http://docs.google.com/java/lang/ClassLoader.html#clearAssertionStatus())()            Sets the default assertion status for this class loader to false and discards any package defaults or class assertion status settings associated with the class loader. |
| protected  [Class](http://docs.google.com/java/lang/Class.html)<?> | [**defineClass**](http://docs.google.com/java/lang/ClassLoader.html#defineClass(byte%5B%5D,%20int,%20int))(byte[] b, int off, int len)  **Deprecated.** *Replaced by* [*defineClass(String, byte[], int, int)*](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20byte%5B%5D,%20int,%20int)) |
| protected  [Class](http://docs.google.com/java/lang/Class.html)<?> | [**defineClass**](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20byte%5B%5D,%20int,%20int))([String](http://docs.google.com/java/lang/String.html) name, byte[] b, int off, int len)            Converts an array of bytes into an instance of class Class. |
| protected  [Class](http://docs.google.com/java/lang/Class.html)<?> | [**defineClass**](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20byte%5B%5D,%20int,%20int,%20java.security.ProtectionDomain))([String](http://docs.google.com/java/lang/String.html) name, byte[] b, int off, int len, [ProtectionDomain](http://docs.google.com/java/security/ProtectionDomain.html) protectionDomain)            Converts an array of bytes into an instance of class Class, with an optional ProtectionDomain. |
| protected  [Class](http://docs.google.com/java/lang/Class.html)<?> | [**defineClass**](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20java.nio.ByteBuffer,%20java.security.ProtectionDomain))([String](http://docs.google.com/java/lang/String.html) name, [ByteBuffer](http://docs.google.com/java/nio/ByteBuffer.html) b, [ProtectionDomain](http://docs.google.com/java/security/ProtectionDomain.html) protectionDomain)            Converts a [ByteBuffer](http://docs.google.com/java/nio/ByteBuffer.html) into an instance of class Class, with an optional ProtectionDomain. |
| protected  [Package](http://docs.google.com/java/lang/Package.html) | [**definePackage**](http://docs.google.com/java/lang/ClassLoader.html#definePackage(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.net.URL))([String](http://docs.google.com/java/lang/String.html) name, [String](http://docs.google.com/java/lang/String.html) specTitle, [String](http://docs.google.com/java/lang/String.html) specVersion, [String](http://docs.google.com/java/lang/String.html) specVendor, [String](http://docs.google.com/java/lang/String.html) implTitle, [String](http://docs.google.com/java/lang/String.html) implVersion, [String](http://docs.google.com/java/lang/String.html) implVendor, [URL](http://docs.google.com/java/net/URL.html) sealBase)            Defines a package by name in this ClassLoader. |
| protected  [Class](http://docs.google.com/java/lang/Class.html)<?> | [**findClass**](http://docs.google.com/java/lang/ClassLoader.html#findClass(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Finds the class with the specified [binary name](#_3znysh7). |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**findLibrary**](http://docs.google.com/java/lang/ClassLoader.html#findLibrary(java.lang.String))([String](http://docs.google.com/java/lang/String.html) libname)            Returns the absolute path name of a native library. |
| protected  [Class](http://docs.google.com/java/lang/Class.html)<?> | [**findLoadedClass**](http://docs.google.com/java/lang/ClassLoader.html#findLoadedClass(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Returns the class with the given [binary name](#_3znysh7) if this loader has been recorded by the Java virtual machine as an initiating loader of a class with that [binary name](#_3znysh7). |
| protected  [URL](http://docs.google.com/java/net/URL.html) | [**findResource**](http://docs.google.com/java/lang/ClassLoader.html#findResource(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Finds the resource with the given name. |
| protected  [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[URL](http://docs.google.com/java/net/URL.html)> | [**findResources**](http://docs.google.com/java/lang/ClassLoader.html#findResources(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Returns an enumeration of [URL](http://docs.google.com/java/net/URL.html) objects representing all the resources with the given name. |
| protected  [Class](http://docs.google.com/java/lang/Class.html)<?> | [**findSystemClass**](http://docs.google.com/java/lang/ClassLoader.html#findSystemClass(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Finds a class with the specified [binary name](#_3znysh7), loading it if necessary. |
| protected  [Package](http://docs.google.com/java/lang/Package.html) | [**getPackage**](http://docs.google.com/java/lang/ClassLoader.html#getPackage(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Returns a Package that has been defined by this class loader or any of its ancestors. |
| protected  [Package](http://docs.google.com/java/lang/Package.html)[] | [**getPackages**](http://docs.google.com/java/lang/ClassLoader.html#getPackages())()            Returns all of the Packages defined by this class loader and its ancestors. |
| [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) | [**getParent**](http://docs.google.com/java/lang/ClassLoader.html#getParent())()            Returns the parent class loader for delegation. |
| [URL](http://docs.google.com/java/net/URL.html) | [**getResource**](http://docs.google.com/java/lang/ClassLoader.html#getResource(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Finds the resource with the given name. |
| [InputStream](http://docs.google.com/java/io/InputStream.html) | [**getResourceAsStream**](http://docs.google.com/java/lang/ClassLoader.html#getResourceAsStream(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Returns an input stream for reading the specified resource. |
| [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[URL](http://docs.google.com/java/net/URL.html)> | [**getResources**](http://docs.google.com/java/lang/ClassLoader.html#getResources(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Finds all the resources with the given name. |
| static [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) | [**getSystemClassLoader**](http://docs.google.com/java/lang/ClassLoader.html#getSystemClassLoader())()            Returns the system class loader for delegation. |
| static [URL](http://docs.google.com/java/net/URL.html) | [**getSystemResource**](http://docs.google.com/java/lang/ClassLoader.html#getSystemResource(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Find a resource of the specified name from the search path used to load classes. |
| static [InputStream](http://docs.google.com/java/io/InputStream.html) | [**getSystemResourceAsStream**](http://docs.google.com/java/lang/ClassLoader.html#getSystemResourceAsStream(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Open for reading, a resource of the specified name from the search path used to load classes. |
| static [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[URL](http://docs.google.com/java/net/URL.html)> | [**getSystemResources**](http://docs.google.com/java/lang/ClassLoader.html#getSystemResources(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Finds all resources of the specified name from the search path used to load classes. |
| [Class](http://docs.google.com/java/lang/Class.html)<?> | [**loadClass**](http://docs.google.com/java/lang/ClassLoader.html#loadClass(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Loads the class with the specified [binary name](#_3znysh7). |
| protected  [Class](http://docs.google.com/java/lang/Class.html)<?> | [**loadClass**](http://docs.google.com/java/lang/ClassLoader.html#loadClass(java.lang.String,%20boolean))([String](http://docs.google.com/java/lang/String.html) name, boolean resolve)            Loads the class with the specified [binary name](#_3znysh7). |
| protected  void | [**resolveClass**](http://docs.google.com/java/lang/ClassLoader.html#resolveClass(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<?> c)            Links the specified class. |
| void | [**setClassAssertionStatus**](http://docs.google.com/java/lang/ClassLoader.html#setClassAssertionStatus(java.lang.String,%20boolean))([String](http://docs.google.com/java/lang/String.html) className, boolean enabled)            Sets the desired assertion status for the named top-level class in this class loader and any nested classes contained therein. |
| void | [**setDefaultAssertionStatus**](http://docs.google.com/java/lang/ClassLoader.html#setDefaultAssertionStatus(boolean))(boolean enabled)            Sets the default assertion status for this class loader. |
| void | [**setPackageAssertionStatus**](http://docs.google.com/java/lang/ClassLoader.html#setPackageAssertionStatus(java.lang.String,%20boolean))([String](http://docs.google.com/java/lang/String.html) packageName, boolean enabled)            Sets the package default assertion status for the named package. |
| protected  void | [**setSigners**](http://docs.google.com/java/lang/ClassLoader.html#setSigners(java.lang.Class,%20java.lang.Object%5B%5D))([Class](http://docs.google.com/java/lang/Class.html)<?> c, [Object](http://docs.google.com/java/lang/Object.html)[] signers)            Sets the signers of a class. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### ClassLoader

protected **ClassLoader**([ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) parent)

Creates a new class loader using the specified parent class loader for delegation.

If there is a security manager, its [checkCreateClassLoader](http://docs.google.com/java/lang/SecurityManager.html#checkCreateClassLoader()) method is invoked. This may result in a security exception.

**Parameters:**parent - The parent class loader **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - If a security manager exists and its checkCreateClassLoader method doesn't allow creation of a new class loader.**Since:** 1.2

### ClassLoader

protected **ClassLoader**()

Creates a new class loader using the ClassLoader returned by the method [getSystemClassLoader()](http://docs.google.com/java/lang/ClassLoader.html#getSystemClassLoader()) as the parent class loader.

If there is a security manager, its [checkCreateClassLoader](http://docs.google.com/java/lang/SecurityManager.html#checkCreateClassLoader()) method is invoked. This may result in a security exception.

**Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - If a security manager exists and its checkCreateClassLoader method doesn't allow creation of a new class loader.

| **Method Detail** |
| --- |

### loadClass

public [Class](http://docs.google.com/java/lang/Class.html)<?> **loadClass**([String](http://docs.google.com/java/lang/String.html) name)  
 throws [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html)

Loads the class with the specified [binary name](#_3znysh7). This method searches for classes in the same manner as the [loadClass(String, boolean)](http://docs.google.com/java/lang/ClassLoader.html#loadClass(java.lang.String,%20boolean)) method. It is invoked by the Java virtual machine to resolve class references. Invoking this method is equivalent to invoking [loadClass(name, false)](http://docs.google.com/java/lang/ClassLoader.html#loadClass(java.lang.String,%20boolean)).

**Parameters:**name - The [binary name](#_3znysh7) of the class **Returns:**The resulting Class object **Throws:** [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html) - If the class was not found

### loadClass

protected [Class](http://docs.google.com/java/lang/Class.html)<?> **loadClass**([String](http://docs.google.com/java/lang/String.html) name,  
 boolean resolve)  
 throws [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html)

Loads the class with the specified [binary name](#_3znysh7). The default implementation of this method searches for classes in the following order:

1. Invoke [findLoadedClass(String)](http://docs.google.com/java/lang/ClassLoader.html#findLoadedClass(java.lang.String)) to check if the class has already been loaded.
2. Invoke the [loadClass](http://docs.google.com/java/lang/ClassLoader.html#loadClass(java.lang.String)) method on the parent class loader. If the parent is null the class loader built-in to the virtual machine is used, instead.
3. Invoke the [findClass(String)](http://docs.google.com/java/lang/ClassLoader.html#findClass(java.lang.String)) method to find the class.

If the class was found using the above steps, and the resolve flag is true, this method will then invoke the [resolveClass(Class)](http://docs.google.com/java/lang/ClassLoader.html#resolveClass(java.lang.Class)) method on the resulting Class object.

Subclasses of ClassLoader are encouraged to override [findClass(String)](http://docs.google.com/java/lang/ClassLoader.html#findClass(java.lang.String)), rather than this method.

**Parameters:**name - The [binary name](#_3znysh7) of the classresolve - If true then resolve the class **Returns:**The resulting Class object **Throws:** [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html) - If the class could not be found

### findClass

protected [Class](http://docs.google.com/java/lang/Class.html)<?> **findClass**([String](http://docs.google.com/java/lang/String.html) name)  
 throws [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html)

Finds the class with the specified [binary name](#_3znysh7). This method should be overridden by class loader implementations that follow the delegation model for loading classes, and will be invoked by the [loadClass](http://docs.google.com/java/lang/ClassLoader.html#loadClass(java.lang.String)) method after checking the parent class loader for the requested class. The default implementation throws a ClassNotFoundException.

**Parameters:**name - The [binary name](#_3znysh7) of the class **Returns:**The resulting Class object **Throws:** [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html) - If the class could not be found**Since:** 1.2

### defineClass

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
protected final [Class](http://docs.google.com/java/lang/Class.html)<?> **defineClass**(byte[] b,  
 int off,  
 int len)  
 throws [ClassFormatError](http://docs.google.com/java/lang/ClassFormatError.html)

**Deprecated.** *Replaced by* [*defineClass(String, byte[], int, int)*](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20byte%5B%5D,%20int,%20int))

Converts an array of bytes into an instance of class Class. Before the Class can be used it must be resolved. This method is deprecated in favor of the version that takes a [binary name](#_3znysh7) as its first argument, and is more secure.

**Parameters:**b - The bytes that make up the class data. The bytes in positions off through off+len-1 should have the format of a valid class file as defined by the [Java Virtual Machine Specification](http://java.sun.com/docs/books/vmspec/).off - The start offset in b of the class datalen - The length of the class data **Returns:**The Class object that was created from the specified class data **Throws:** [ClassFormatError](http://docs.google.com/java/lang/ClassFormatError.html) - If the data did not contain a valid class [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - If either off or len is negative, or if off+len is greater than b.length.**See Also:**[loadClass(String, boolean)](http://docs.google.com/java/lang/ClassLoader.html#loadClass(java.lang.String,%20boolean)), [resolveClass(Class)](http://docs.google.com/java/lang/ClassLoader.html#resolveClass(java.lang.Class))

### defineClass

protected final [Class](http://docs.google.com/java/lang/Class.html)<?> **defineClass**([String](http://docs.google.com/java/lang/String.html) name,  
 byte[] b,  
 int off,  
 int len)  
 throws [ClassFormatError](http://docs.google.com/java/lang/ClassFormatError.html)

Converts an array of bytes into an instance of class Class. Before the Class can be used it must be resolved.

This method assigns a default [ProtectionDomain](http://docs.google.com/java/security/ProtectionDomain.html) to the newly defined class. The ProtectionDomain is effectively granted the same set of permissions returned when [Policy.getPolicy().getPermissions(new CodeSource(null, null))](http://docs.google.com/java/security/Policy.html#getPermissions(java.security.CodeSource)) is invoked. The default domain is created on the first invocation of [defineClass](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20byte%5B%5D,%20int,%20int)), and re-used on subsequent invocations.

To assign a specific ProtectionDomain to the class, use the [defineClass](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20byte%5B%5D,%20int,%20int,%20java.security.ProtectionDomain)) method that takes a ProtectionDomain as one of its arguments.

**Parameters:**name - The expected [binary name](#_3znysh7) of the class, or null if not knownb - The bytes that make up the class data. The bytes in positions off through off+len-1 should have the format of a valid class file as defined by the [Java Virtual Machine Specification](http://java.sun.com/docs/books/vmspec/).off - The start offset in b of the class datalen - The length of the class data **Returns:**The Class object that was created from the specified class data. **Throws:** [ClassFormatError](http://docs.google.com/java/lang/ClassFormatError.html) - If the data did not contain a valid class [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - If either off or len is negative, or if off+len is greater than b.length. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - If an attempt is made to add this class to a package that contains classes that were signed by a different set of certificates than this class (which is unsigned), or if name begins with "java.".**Since:** 1.1 **See Also:**[loadClass(String, boolean)](http://docs.google.com/java/lang/ClassLoader.html#loadClass(java.lang.String,%20boolean)), [resolveClass(Class)](http://docs.google.com/java/lang/ClassLoader.html#resolveClass(java.lang.Class)), [CodeSource](http://docs.google.com/java/security/CodeSource.html), [SecureClassLoader](http://docs.google.com/java/security/SecureClassLoader.html)

### defineClass

protected final [Class](http://docs.google.com/java/lang/Class.html)<?> **defineClass**([String](http://docs.google.com/java/lang/String.html) name,  
 byte[] b,  
 int off,  
 int len,  
 [ProtectionDomain](http://docs.google.com/java/security/ProtectionDomain.html) protectionDomain)  
 throws [ClassFormatError](http://docs.google.com/java/lang/ClassFormatError.html)

Converts an array of bytes into an instance of class Class, with an optional ProtectionDomain. If the domain is null, then a default domain will be assigned to the class as specified in the documentation for [defineClass(String, byte[], int, int)](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20byte%5B%5D,%20int,%20int)). Before the class can be used it must be resolved.

The first class defined in a package determines the exact set of certificates that all subsequent classes defined in that package must contain. The set of certificates for a class is obtained from the [CodeSource](http://docs.google.com/java/security/CodeSource.html) within the ProtectionDomain of the class. Any classes added to that package must contain the same set of certificates or a SecurityException will be thrown. Note that if name is null, this check is not performed. You should always pass in the [binary name](#_3znysh7) of the class you are defining as well as the bytes. This ensures that the class you are defining is indeed the class you think it is.

The specified name cannot begin with "java.", since all classes in the "java.\* packages can only be defined by the bootstrap class loader. If name is not null, it must be equal to the [binary name](#_3znysh7) of the class specified by the byte array "b", otherwise a NoClassDefFoundError will be thrown.

**Parameters:**name - The expected [binary name](#_3znysh7) of the class, or null if not knownb - The bytes that make up the class data. The bytes in positions off through off+len-1 should have the format of a valid class file as defined by the [Java Virtual Machine Specification](http://java.sun.com/docs/books/vmspec/).off - The start offset in b of the class datalen - The length of the class dataprotectionDomain - The ProtectionDomain of the class **Returns:**The Class object created from the data, and optional ProtectionDomain. **Throws:** [ClassFormatError](http://docs.google.com/java/lang/ClassFormatError.html) - If the data did not contain a valid class [NoClassDefFoundError](http://docs.google.com/java/lang/NoClassDefFoundError.html) - If name is not equal to the [binary name](#_3znysh7) of the class specified by b [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - If either off or len is negative, or if off+len is greater than b.length. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - If an attempt is made to add this class to a package that contains classes that were signed by a different set of certificates than this class, or if name begins with "java.".

### defineClass

protected final [Class](http://docs.google.com/java/lang/Class.html)<?> **defineClass**([String](http://docs.google.com/java/lang/String.html) name,  
 [ByteBuffer](http://docs.google.com/java/nio/ByteBuffer.html) b,  
 [ProtectionDomain](http://docs.google.com/java/security/ProtectionDomain.html) protectionDomain)  
 throws [ClassFormatError](http://docs.google.com/java/lang/ClassFormatError.html)

Converts a [ByteBuffer](http://docs.google.com/java/nio/ByteBuffer.html) into an instance of class Class, with an optional ProtectionDomain. If the domain is null, then a default domain will be assigned to the class as specified in the documentation for [defineClass(String, byte[], int, int)](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20byte%5B%5D,%20int,%20int)). Before the class can be used it must be resolved.

The rules about the first class defined in a package determining the set of certificates for the package, and the restrictions on class names are identical to those specified in the documentation for [defineClass(String, byte[], int, int, ProtectionDomain)](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20byte%5B%5D,%20int,%20int,%20java.security.ProtectionDomain)).

An invocation of this method of the form *cl*.defineClass(*name*, *bBuffer*, *pd*) yields exactly the same result as the statements

...

byte[] temp = new byte[*bBuffer*.[remaining](http://docs.google.com/java/nio/Buffer.html#remaining())()];

*bBuffer*.[get](http://docs.google.com/java/nio/ByteBuffer.html#get(byte%5B%5D))(temp);

return [*cl*.defineClass](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20byte%5B%5D,%20int,%20int,%20java.security.ProtectionDomain))(*name*, temp, 0, temp.length, *pd*);

**Parameters:**name - The expected [binary namenull if not knownb - The bytes that make up the class data. The bytes from positions b.position() through b.position() + b.limit() -1 should have the format of a valid class file as defined by the](#_3znysh7) [Java Virtual Machine Specification](http://java.sun.com/docs/books/vmspec/).protectionDomain - The ProtectionDomain of the class, or null. **Returns:**The Class object created from the data, and optional ProtectionDomain. **Throws:** [ClassFormatError](http://docs.google.com/java/lang/ClassFormatError.html) - If the data did not contain a valid class. [NoClassDefFoundError](http://docs.google.com/java/lang/NoClassDefFoundError.html) - If name is not equal to the [binary name](#_3znysh7) of the class specified by b [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - If an attempt is made to add this class to a package that contains classes that were signed by a different set of certificates than this class, or if name begins with "java.".**Since:** 1.5 **See Also:**[defineClass(String, byte[], int, int, ProtectionDomain)](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20byte%5B%5D,%20int,%20int,%20java.security.ProtectionDomain))

### resolveClass

protected final void **resolveClass**([Class](http://docs.google.com/java/lang/Class.html)<?> c)

Links the specified class. This (misleadingly named) method may be used by a class loader to link a class. If the class c has already been linked, then this method simply returns. Otherwise, the class is linked as described in the "Execution" chapter of the [Java Language Specification](http://java.sun.com/docs/books/jls/).

**Parameters:**c - The class to link **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - If c is null.**See Also:**[defineClass(String, byte[], int, int)](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20byte%5B%5D,%20int,%20int))

### findSystemClass

protected final [Class](http://docs.google.com/java/lang/Class.html)<?> **findSystemClass**([String](http://docs.google.com/java/lang/String.html) name)  
 throws [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html)

Finds a class with the specified [binary name](#_3znysh7), loading it if necessary.

This method loads the class through the system class loader (see [getSystemClassLoader()](http://docs.google.com/java/lang/ClassLoader.html#getSystemClassLoader())). The Class object returned might have more than one ClassLoader associated with it. Subclasses of ClassLoader need not usually invoke this method, because most class loaders need to override just [findClass(String)](http://docs.google.com/java/lang/ClassLoader.html#findClass(java.lang.String)).

**Parameters:**name - The [binary name](#_3znysh7) of the class **Returns:**The Class object for the specified name **Throws:** [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html) - If the class could not be found**See Also:**[ClassLoader(ClassLoader)](http://docs.google.com/java/lang/ClassLoader.html#ClassLoader(java.lang.ClassLoader)), [getParent()](http://docs.google.com/java/lang/ClassLoader.html#getParent())

### findLoadedClass

protected final [Class](http://docs.google.com/java/lang/Class.html)<?> **findLoadedClass**([String](http://docs.google.com/java/lang/String.html) name)

Returns the class with the given [binary name](#_3znysh7) if this loader has been recorded by the Java virtual machine as an initiating loader of a class with that [binary name](#_3znysh7). Otherwise null is returned.

**Parameters:**name - The [binary name](#_3znysh7) of the class **Returns:**The Class object, or null if the class has not been loaded**Since:** 1.1

### setSigners

protected final void **setSigners**([Class](http://docs.google.com/java/lang/Class.html)<?> c,  
 [Object](http://docs.google.com/java/lang/Object.html)[] signers)

Sets the signers of a class. This should be invoked after defining a class.

**Parameters:**c - The Class objectsigners - The signers for the class**Since:** 1.1

### getResource

public [URL](http://docs.google.com/java/net/URL.html) **getResource**([String](http://docs.google.com/java/lang/String.html) name)

Finds the resource with the given name. A resource is some data (images, audio, text, etc) that can be accessed by class code in a way that is independent of the location of the code.

The name of a resource is a '/'-separated path name that identifies the resource.

This method will first search the parent class loader for the resource; if the parent is null the path of the class loader built-in to the virtual machine is searched. That failing, this method will invoke [findResource(String)](http://docs.google.com/java/lang/ClassLoader.html#findResource(java.lang.String)) to find the resource.

**Parameters:**name - The resource name **Returns:**A URL object for reading the resource, or null if the resource could not be found or the invoker doesn't have adequate privileges to get the resource.**Since:** 1.1

### getResources

public [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[URL](http://docs.google.com/java/net/URL.html)> **getResources**([String](http://docs.google.com/java/lang/String.html) name)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Finds all the resources with the given name. A resource is some data (images, audio, text, etc) that can be accessed by class code in a way that is independent of the location of the code.

The name of a resource is a /-separated path name that identifies the resource.

The search order is described in the documentation for [getResource(String)](http://docs.google.com/java/lang/ClassLoader.html#getResource(java.lang.String)).

**Parameters:**name - The resource name **Returns:**An enumeration of [URL](http://docs.google.com/java/net/URL.html) objects for the resource. If no resources could be found, the enumeration will be empty. Resources that the class loader doesn't have access to will not be in the enumeration. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If I/O errors occur**Since:** 1.2 **See Also:**[findResources(String)](http://docs.google.com/java/lang/ClassLoader.html#findResources(java.lang.String))

### findResource

protected [URL](http://docs.google.com/java/net/URL.html) **findResource**([String](http://docs.google.com/java/lang/String.html) name)

Finds the resource with the given name. Class loader implementations should override this method to specify where to find resources.

**Parameters:**name - The resource name **Returns:**A URL object for reading the resource, or null if the resource could not be found**Since:** 1.2

### findResources

protected [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[URL](http://docs.google.com/java/net/URL.html)> **findResources**([String](http://docs.google.com/java/lang/String.html) name)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Returns an enumeration of [URL](http://docs.google.com/java/net/URL.html) objects representing all the resources with the given name. Class loader implementations should override this method to specify where to load resources from.

**Parameters:**name - The resource name **Returns:**An enumeration of [URL](http://docs.google.com/java/net/URL.html) objects for the resources **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If I/O errors occur**Since:** 1.2

### getSystemResource

public static [URL](http://docs.google.com/java/net/URL.html) **getSystemResource**([String](http://docs.google.com/java/lang/String.html) name)

Find a resource of the specified name from the search path used to load classes. This method locates the resource through the system class loader (see [getSystemClassLoader()](http://docs.google.com/java/lang/ClassLoader.html#getSystemClassLoader())).

**Parameters:**name - The resource name **Returns:**A [URL](http://docs.google.com/java/net/URL.html) object for reading the resource, or null if the resource could not be found**Since:** 1.1

### getSystemResources

public static [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[URL](http://docs.google.com/java/net/URL.html)> **getSystemResources**([String](http://docs.google.com/java/lang/String.html) name)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Finds all resources of the specified name from the search path used to load classes. The resources thus found are returned as an [Enumeration](http://docs.google.com/java/util/Enumeration.html) of [URL](http://docs.google.com/java/net/URL.html) objects.

The search order is described in the documentation for [getSystemResource(String)](http://docs.google.com/java/lang/ClassLoader.html#getSystemResource(java.lang.String)).

**Parameters:**name - The resource name **Returns:**An enumeration of resource [URL](http://docs.google.com/java/net/URL.html) objects **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If I/O errors occur**Since:** 1.2

### getResourceAsStream

public [InputStream](http://docs.google.com/java/io/InputStream.html) **getResourceAsStream**([String](http://docs.google.com/java/lang/String.html) name)

Returns an input stream for reading the specified resource.

The search order is described in the documentation for [getResource(String)](http://docs.google.com/java/lang/ClassLoader.html#getResource(java.lang.String)).

**Parameters:**name - The resource name **Returns:**An input stream for reading the resource, or null if the resource could not be found**Since:** 1.1

### getSystemResourceAsStream

public static [InputStream](http://docs.google.com/java/io/InputStream.html) **getSystemResourceAsStream**([String](http://docs.google.com/java/lang/String.html) name)

Open for reading, a resource of the specified name from the search path used to load classes. This method locates the resource through the system class loader (see [getSystemClassLoader()](http://docs.google.com/java/lang/ClassLoader.html#getSystemClassLoader())).

**Parameters:**name - The resource name **Returns:**An input stream for reading the resource, or null if the resource could not be found**Since:** 1.1

### getParent

public final [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) **getParent**()

Returns the parent class loader for delegation. Some implementations may use null to represent the bootstrap class loader. This method will return null in such implementations if this class loader's parent is the bootstrap class loader.

If a security manager is present, and the invoker's class loader is not null and is not an ancestor of this class loader, then this method invokes the security manager's [checkPermission](http://docs.google.com/java/lang/SecurityManager.html#checkPermission(java.security.Permission)) method with a [RuntimePermission("getClassLoader")](http://docs.google.com/java/lang/RuntimePermission.html#RuntimePermission(java.lang.String)) permission to verify access to the parent class loader is permitted. If not, a SecurityException will be thrown.

**Returns:**The parent ClassLoader **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - If a security manager exists and its checkPermission method doesn't allow access to this class loader's parent class loader.**Since:** 1.2

### getSystemClassLoader

public static [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) **getSystemClassLoader**()

Returns the system class loader for delegation. This is the default delegation parent for new ClassLoader instances, and is typically the class loader used to start the application.

This method is first invoked early in the runtime's startup sequence, at which point it creates the system class loader and sets it as the context class loader of the invoking Thread.

The default system class loader is an implementation-dependent instance of this class.

If the system property "java.system.class.loader" is defined when this method is first invoked then the value of that property is taken to be the name of a class that will be returned as the system class loader. The class is loaded using the default system class loader and must define a public constructor that takes a single parameter of type ClassLoader which is used as the delegation parent. An instance is then created using this constructor with the default system class loader as the parameter. The resulting class loader is defined to be the system class loader.

If a security manager is present, and the invoker's class loader is not null and the invoker's class loader is not the same as or an ancestor of the system class loader, then this method invokes the security manager's [checkPermission](http://docs.google.com/java/lang/SecurityManager.html#checkPermission(java.security.Permission)) method with a [RuntimePermission("getClassLoader")](http://docs.google.com/java/lang/RuntimePermission.html#RuntimePermission(java.lang.String)) permission to verify access to the system class loader. If not, a SecurityException will be thrown.

**Returns:**The system ClassLoader for delegation, or null if none **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - If a security manager exists and its checkPermission method doesn't allow access to the system class loader. [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - If invoked recursively during the construction of the class loader specified by the "java.system.class.loader" property. [Error](http://docs.google.com/java/lang/Error.html) - If the system property "java.system.class.loader" is defined but the named class could not be loaded, the provider class does not define the required constructor, or an exception is thrown by that constructor when it is invoked. The underlying cause of the error can be retrieved via the [Throwable.getCause()](http://docs.google.com/java/lang/Throwable.html#getCause()) method.

### definePackage

protected [Package](http://docs.google.com/java/lang/Package.html) **definePackage**([String](http://docs.google.com/java/lang/String.html) name,  
 [String](http://docs.google.com/java/lang/String.html) specTitle,  
 [String](http://docs.google.com/java/lang/String.html) specVersion,  
 [String](http://docs.google.com/java/lang/String.html) specVendor,  
 [String](http://docs.google.com/java/lang/String.html) implTitle,  
 [String](http://docs.google.com/java/lang/String.html) implVersion,  
 [String](http://docs.google.com/java/lang/String.html) implVendor,  
 [URL](http://docs.google.com/java/net/URL.html) sealBase)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html)

Defines a package by name in this ClassLoader. This allows class loaders to define the packages for their classes. Packages must be created before the class is defined, and package names must be unique within a class loader and cannot be redefined or changed once created.

**Parameters:**name - The package namespecTitle - The specification titlespecVersion - The specification versionspecVendor - The specification vendorimplTitle - The implementation titleimplVersion - The implementation versionimplVendor - The implementation vendorsealBase - If not null, then this package is sealed with respect to the given code source [URL](http://docs.google.com/java/net/URL.html) object. Otherwise, the package is not sealed. **Returns:**The newly defined Package object **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If package name duplicates an existing package either in this class loader or one of its ancestors**Since:** 1.2

### getPackage

protected [Package](http://docs.google.com/java/lang/Package.html) **getPackage**([String](http://docs.google.com/java/lang/String.html) name)

Returns a Package that has been defined by this class loader or any of its ancestors.

**Parameters:**name - The package name **Returns:**The Package corresponding to the given name, or null if not found**Since:** 1.2

### getPackages

protected [Package](http://docs.google.com/java/lang/Package.html)[] **getPackages**()

Returns all of the Packages defined by this class loader and its ancestors.

**Returns:**The array of Package objects defined by this ClassLoader**Since:** 1.2

### findLibrary

protected [String](http://docs.google.com/java/lang/String.html) **findLibrary**([String](http://docs.google.com/java/lang/String.html) libname)

Returns the absolute path name of a native library. The VM invokes this method to locate the native libraries that belong to classes loaded with this class loader. If this method returns null, the VM searches the library along the path specified as the "java.library.path" property.

**Parameters:**libname - The library name **Returns:**The absolute path of the native library**Since:** 1.2 **See Also:**[System.loadLibrary(String)](http://docs.google.com/java/lang/System.html#loadLibrary(java.lang.String)), [System.mapLibraryName(String)](http://docs.google.com/java/lang/System.html#mapLibraryName(java.lang.String))

### setDefaultAssertionStatus

public void **setDefaultAssertionStatus**(boolean enabled)

Sets the default assertion status for this class loader. This setting determines whether classes loaded by this class loader and initialized in the future will have assertions enabled or disabled by default. This setting may be overridden on a per-package or per-class basis by invoking [setPackageAssertionStatus(String, boolean)](http://docs.google.com/java/lang/ClassLoader.html#setPackageAssertionStatus(java.lang.String,%20boolean)) or [setClassAssertionStatus(String, boolean)](http://docs.google.com/java/lang/ClassLoader.html#setClassAssertionStatus(java.lang.String,%20boolean)).

**Parameters:**enabled - true if classes loaded by this class loader will henceforth have assertions enabled by default, false if they will have assertions disabled by default.**Since:** 1.4

### setPackageAssertionStatus

public void **setPackageAssertionStatus**([String](http://docs.google.com/java/lang/String.html) packageName,  
 boolean enabled)

Sets the package default assertion status for the named package. The package default assertion status determines the assertion status for classes initialized in the future that belong to the named package or any of its "subpackages".

A subpackage of a package named p is any package whose name begins with "p.". For example, javax.swing.text is a subpackage of javax.swing, and both java.util and java.lang.reflect are subpackages of java.

In the event that multiple package defaults apply to a given class, the package default pertaining to the most specific package takes precedence over the others. For example, if javax.lang and javax.lang.reflect both have package defaults associated with them, the latter package default applies to classes in javax.lang.reflect.

Package defaults take precedence over the class loader's default assertion status, and may be overridden on a per-class basis by invoking [setClassAssertionStatus(String, boolean)](http://docs.google.com/java/lang/ClassLoader.html#setClassAssertionStatus(java.lang.String,%20boolean)).

**Parameters:**packageName - The name of the package whose package default assertion status is to be set. A null value indicates the unnamed package that is "current" ([Java Language Specification](http://java.sun.com/docs/books/jls/), section 7.4.2).enabled - true if classes loaded by this classloader and belonging to the named package or any of its subpackages will have assertions enabled by default, false if they will have assertions disabled by default.**Since:** 1.4

### setClassAssertionStatus

public void **setClassAssertionStatus**([String](http://docs.google.com/java/lang/String.html) className,  
 boolean enabled)

Sets the desired assertion status for the named top-level class in this class loader and any nested classes contained therein. This setting takes precedence over the class loader's default assertion status, and over any applicable per-package default. This method has no effect if the named class has already been initialized. (Once a class is initialized, its assertion status cannot change.)

If the named class is not a top-level class, this invocation will have no effect on the actual assertion status of any class.

**Parameters:**className - The fully qualified class name of the top-level class whose assertion status is to be set.enabled - true if the named class is to have assertions enabled when (and if) it is initialized, false if the class is to have assertions disabled.**Since:** 1.4

### clearAssertionStatus

public void **clearAssertionStatus**()

Sets the default assertion status for this class loader to false and discards any package defaults or class assertion status settings associated with the class loader. This method is provided so that class loaders can be made to ignore any command line or persistent assertion status settings and "start with a clean slate."

**Since:** 1.4

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ClassLoader.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/lang/ClassFormatError.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/ClassNotFoundException.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/ClassLoader.html)    [**NO FRAMES**](http://docs.google.com/ClassLoader.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#17dp8vu) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).