| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DriverManager.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/sql/Driver.html)   [**NEXT CLASS**](http://docs.google.com/java/sql/DriverPropertyInfo.html) | [**FRAMES**](http://docs.google.com/index.html?java/sql/DriverManager.html)    [**NO FRAMES**](http://docs.google.com/DriverManager.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

## **java.sql**

Class DriverManager

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

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

The basic service for managing a set of JDBC drivers.

**NOTE:** The DataSource interface, new in the JDBC 2.0 API, provides another way to connect to a data source. The use of a DataSource object is the preferred means of connecting to a data source.

As part of its initialization, the DriverManager class will attempt to load the driver classes referenced in the "jdbc.drivers" system property. This allows a user to customize the JDBC Drivers used by their applications. For example in your ~/.hotjava/properties file you might specify:

jdbc.drivers=foo.bah.Driver:wombat.sql.Driver:bad.taste.ourDriver

The DriverManager methods getConnection and getDrivers have been enhanced to support the Java Standard Edition [Service Provider](http://docs.google.com/technotes/guides/jar/jar.html#Service%20Provider) mechanism. JDBC 4.0 Drivers must include the file META-INF/services/java.sql.Driver. This file contains the name of the JDBC drivers implementation of java.sql.Driver. For example, to load the my.sql.Driver class, the META-INF/services/java.sql.Driver file would contain the entry:

my.sql.Driver

Applications no longer need to explictly load JDBC drivers using Class.forName(). Existing programs which currently load JDBC drivers using Class.forName() will continue to work without modification.

When the method getConnection is called, the DriverManager will attempt to locate a suitable driver from amongst those loaded at initialization and those loaded explicitly using the same classloader as the current applet or application.

Starting with the Java 2 SDK, Standard Edition, version 1.3, a logging stream can be set only if the proper permission has been granted. Normally this will be done with the tool PolicyTool, which can be used to grant permission java.sql.SQLPermission "setLog".

**See Also:**[Driver](http://docs.google.com/java/sql/Driver.html), [Connection](http://docs.google.com/java/sql/Connection.html)

| **Method Summary** | |
| --- | --- |
| static void | [**deregisterDriver**](http://docs.google.com/java/sql/DriverManager.html#deregisterDriver(java.sql.Driver))([Driver](http://docs.google.com/java/sql/Driver.html) driver)            Drops a driver from the DriverManager's list. |
| static [Connection](http://docs.google.com/java/sql/Connection.html) | [**getConnection**](http://docs.google.com/java/sql/DriverManager.html#getConnection(java.lang.String))([String](http://docs.google.com/java/lang/String.html) url)            Attempts to establish a connection to the given database URL. |
| static [Connection](http://docs.google.com/java/sql/Connection.html) | [**getConnection**](http://docs.google.com/java/sql/DriverManager.html#getConnection(java.lang.String,%20java.util.Properties))([String](http://docs.google.com/java/lang/String.html) url, [Properties](http://docs.google.com/java/util/Properties.html) info)            Attempts to establish a connection to the given database URL. |
| static [Connection](http://docs.google.com/java/sql/Connection.html) | [**getConnection**](http://docs.google.com/java/sql/DriverManager.html#getConnection(java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) url, [String](http://docs.google.com/java/lang/String.html) user, [String](http://docs.google.com/java/lang/String.html) password)            Attempts to establish a connection to the given database URL. |
| static [Driver](http://docs.google.com/java/sql/Driver.html) | [**getDriver**](http://docs.google.com/java/sql/DriverManager.html#getDriver(java.lang.String))([String](http://docs.google.com/java/lang/String.html) url)            Attempts to locate a driver that understands the given URL. |
| static [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[Driver](http://docs.google.com/java/sql/Driver.html)> | [**getDrivers**](http://docs.google.com/java/sql/DriverManager.html#getDrivers())()            Retrieves an Enumeration with all of the currently loaded JDBC drivers to which the current caller has access. |
| static int | [**getLoginTimeout**](http://docs.google.com/java/sql/DriverManager.html#getLoginTimeout())()            Gets the maximum time in seconds that a driver can wait when attempting to log in to a database. |
| static [PrintStream](http://docs.google.com/java/io/PrintStream.html) | [**getLogStream**](http://docs.google.com/java/sql/DriverManager.html#getLogStream())()  **Deprecated.** |
| static [PrintWriter](http://docs.google.com/java/io/PrintWriter.html) | [**getLogWriter**](http://docs.google.com/java/sql/DriverManager.html#getLogWriter())()            Retrieves the log writer. |
| static void | [**println**](http://docs.google.com/java/sql/DriverManager.html#println(java.lang.String))([String](http://docs.google.com/java/lang/String.html) message)            Prints a message to the current JDBC log stream. |
| static void | [**registerDriver**](http://docs.google.com/java/sql/DriverManager.html#registerDriver(java.sql.Driver))([Driver](http://docs.google.com/java/sql/Driver.html) driver)            Registers the given driver with the DriverManager. |
| static void | [**setLoginTimeout**](http://docs.google.com/java/sql/DriverManager.html#setLoginTimeout(int))(int seconds)            Sets the maximum time in seconds that a driver will wait while attempting to connect to a database. |
| static void | [**setLogStream**](http://docs.google.com/java/sql/DriverManager.html#setLogStream(java.io.PrintStream))([PrintStream](http://docs.google.com/java/io/PrintStream.html) out)  **Deprecated.** |
| static void | [**setLogWriter**](http://docs.google.com/java/sql/DriverManager.html#setLogWriter(java.io.PrintWriter))([PrintWriter](http://docs.google.com/java/io/PrintWriter.html) out)            Sets the logging/tracing PrintWriter object that is used by the DriverManager and all drivers. |

| **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)) |

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

### getLogWriter

public static [PrintWriter](http://docs.google.com/java/io/PrintWriter.html) **getLogWriter**()

Retrieves the log writer. The getLogWriter and setLogWriter methods should be used instead of the get/setlogStream methods, which are deprecated.

**Returns:**a java.io.PrintWriter object**Since:** 1.2 **See Also:**[setLogWriter(java.io.PrintWriter)](http://docs.google.com/java/sql/DriverManager.html#setLogWriter(java.io.PrintWriter))

### setLogWriter

public static void **setLogWriter**([PrintWriter](http://docs.google.com/java/io/PrintWriter.html) out)

Sets the logging/tracing PrintWriter object that is used by the DriverManager and all drivers.

There is a minor versioning problem created by the introduction of the method setLogWriter. The method setLogWriter cannot create a PrintStream object that will be returned by getLogStream---the Java platform does not provide a backward conversion. As a result, a new application that uses setLogWriter and also uses a JDBC 1.0 driver that uses getLogStream will likely not see debugging information written by that driver.

Starting with the Java 2 SDK, Standard Edition, version 1.3 release, this method checks to see that there is an SQLPermission object before setting the logging stream. If a SecurityManager exists and its checkPermission method denies setting the log writer, this method throws a java.lang.SecurityException.

**Parameters:**out - the new logging/tracing PrintStream object; null to disable logging and tracing **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkPermission method denies setting the log writer**Since:** 1.2 **See Also:**[SecurityManager.checkPermission(java.security.Permission)](http://docs.google.com/java/lang/SecurityManager.html#checkPermission(java.security.Permission)), [getLogWriter()](http://docs.google.com/java/sql/DriverManager.html#getLogWriter())

### getConnection

public static [Connection](http://docs.google.com/java/sql/Connection.html) **getConnection**([String](http://docs.google.com/java/lang/String.html) url,  
 [Properties](http://docs.google.com/java/util/Properties.html) info)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Attempts to establish a connection to the given database URL. The DriverManager attempts to select an appropriate driver from the set of registered JDBC drivers.

**Parameters:**url - a database url of the form jdbc:*subprotocol*:*subname*info - a list of arbitrary string tag/value pairs as connection arguments; normally at least a "user" and "password" property should be included **Returns:**a Connection to the URL **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### getConnection

public static [Connection](http://docs.google.com/java/sql/Connection.html) **getConnection**([String](http://docs.google.com/java/lang/String.html) url,  
 [String](http://docs.google.com/java/lang/String.html) user,  
 [String](http://docs.google.com/java/lang/String.html) password)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Attempts to establish a connection to the given database URL. The DriverManager attempts to select an appropriate driver from the set of registered JDBC drivers.

**Parameters:**url - a database url of the form jdbc:*subprotocol*:*subname*user - the database user on whose behalf the connection is being madepassword - the user's password **Returns:**a connection to the URL **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### getConnection

public static [Connection](http://docs.google.com/java/sql/Connection.html) **getConnection**([String](http://docs.google.com/java/lang/String.html) url)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Attempts to establish a connection to the given database URL. The DriverManager attempts to select an appropriate driver from the set of registered JDBC drivers.

**Parameters:**url - a database url of the form jdbc:*subprotocol*:*subname* **Returns:**a connection to the URL **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### getDriver

public static [Driver](http://docs.google.com/java/sql/Driver.html) **getDriver**([String](http://docs.google.com/java/lang/String.html) url)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Attempts to locate a driver that understands the given URL. The DriverManager attempts to select an appropriate driver from the set of registered JDBC drivers.

**Parameters:**url - a database URL of the form jdbc:*subprotocol*:*subname* **Returns:**a Driver object representing a driver that can connect to the given URL **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### registerDriver

public static void **registerDriver**([Driver](http://docs.google.com/java/sql/Driver.html) driver)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Registers the given driver with the DriverManager. A newly-loaded driver class should call the method registerDriver to make itself known to the DriverManager.

**Parameters:**driver - the new JDBC Driver that is to be registered with the DriverManager **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### deregisterDriver

public static void **deregisterDriver**([Driver](http://docs.google.com/java/sql/Driver.html) driver)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Drops a driver from the DriverManager's list. Applets can only deregister drivers from their own classloaders.

**Parameters:**driver - the JDBC Driver to drop **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### getDrivers

public static [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[Driver](http://docs.google.com/java/sql/Driver.html)> **getDrivers**()

Retrieves an Enumeration with all of the currently loaded JDBC drivers to which the current caller has access.

**Note:** The classname of a driver can be found using d.getClass().getName()

**Returns:**the list of JDBC Drivers loaded by the caller's class loader

### setLoginTimeout

public static void **setLoginTimeout**(int seconds)

Sets the maximum time in seconds that a driver will wait while attempting to connect to a database.

**Parameters:**seconds - the login time limit in seconds; zero means there is no limit**See Also:**[getLoginTimeout()](http://docs.google.com/java/sql/DriverManager.html#getLoginTimeout())

### getLoginTimeout

public static int **getLoginTimeout**()

Gets the maximum time in seconds that a driver can wait when attempting to log in to a database.

**Returns:**the driver login time limit in seconds**See Also:**[setLoginTimeout(int)](http://docs.google.com/java/sql/DriverManager.html#setLoginTimeout(int))

### setLogStream

public static void **setLogStream**([PrintStream](http://docs.google.com/java/io/PrintStream.html) out)

**Deprecated.**

Sets the logging/tracing PrintStream that is used by the DriverManager and all drivers.

In the Java 2 SDK, Standard Edition, version 1.3 release, this method checks to see that there is an SQLPermission object before setting the logging stream. If a SecurityManager exists and its checkPermission method denies setting the log writer, this method throws a java.lang.SecurityException.

**Parameters:**out - the new logging/tracing PrintStream; to disable, set to null **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkPermission method denies setting the log stream**See Also:**[SecurityManager.checkPermission(java.security.Permission)](http://docs.google.com/java/lang/SecurityManager.html#checkPermission(java.security.Permission)), [getLogStream()](http://docs.google.com/java/sql/DriverManager.html#getLogStream())

### getLogStream

public static [PrintStream](http://docs.google.com/java/io/PrintStream.html) **getLogStream**()

**Deprecated.**

Retrieves the logging/tracing PrintStream that is used by the DriverManager and all drivers.

**Returns:**the logging/tracing PrintStream; if disabled, is null**See Also:**[setLogStream(java.io.PrintStream)](http://docs.google.com/java/sql/DriverManager.html#setLogStream(java.io.PrintStream))

### println

public static void **println**([String](http://docs.google.com/java/lang/String.html) message)

Prints a message to the current JDBC log stream.

**Parameters:**message - a log or tracing message

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DriverManager.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/sql/Driver.html)   [**NEXT CLASS**](http://docs.google.com/java/sql/DriverPropertyInfo.html) | [**FRAMES**](http://docs.google.com/index.html?java/sql/DriverManager.html)    [**NO FRAMES**](http://docs.google.com/DriverManager.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

[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).