| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Driver.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/Date.html)   [**NEXT CLASS**](http://docs.google.com/java/sql/DriverManager.html) | [**FRAMES**](http://docs.google.com/index.html?java/sql/Driver.html)    [**NO FRAMES**](http://docs.google.com/Driver.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

## **java.sql**

Interface Driver

public interface **Driver**

The interface that every driver class must implement.

The Java SQL framework allows for multiple database drivers.

Each driver should supply a class that implements the Driver interface.

The DriverManager will try to load as many drivers as it can find and then for any given connection request, it will ask each driver in turn to try to connect to the target URL.

It is strongly recommended that each Driver class should be small and standalone so that the Driver class can be loaded and queried without bringing in vast quantities of supporting code.

When a Driver class is loaded, it should create an instance of itself and register it with the DriverManager. This means that a user can load and register a driver by calling

Class.forName("foo.bah.Driver")

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

| **Method Summary** | |
| --- | --- |
| boolean | [**acceptsURL**](http://docs.google.com/java/sql/Driver.html#acceptsURL(java.lang.String))([String](http://docs.google.com/java/lang/String.html) url)            Retrieves whether the driver thinks that it can open a connection to the given URL. |
| [Connection](http://docs.google.com/java/sql/Connection.html) | [**connect**](http://docs.google.com/java/sql/Driver.html#connect(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 make a database connection to the given URL. |
| int | [**getMajorVersion**](http://docs.google.com/java/sql/Driver.html#getMajorVersion())()            Retrieves the driver's major version number. |
| int | [**getMinorVersion**](http://docs.google.com/java/sql/Driver.html#getMinorVersion())()            Gets the driver's minor version number. |
| [DriverPropertyInfo](http://docs.google.com/java/sql/DriverPropertyInfo.html)[] | [**getPropertyInfo**](http://docs.google.com/java/sql/Driver.html#getPropertyInfo(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)            Gets information about the possible properties for this driver. |
| boolean | [**jdbcCompliant**](http://docs.google.com/java/sql/Driver.html#jdbcCompliant())()            Reports whether this driver is a genuine JDBC CompliantTM driver. |

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

### connect

[Connection](http://docs.google.com/java/sql/Connection.html) **connect**([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 make a database connection to the given URL. The driver should return "null" if it realizes it is the wrong kind of driver to connect to the given URL. This will be common, as when the JDBC driver manager is asked to connect to a given URL it passes the URL to each loaded driver in turn.

The driver should throw an SQLException if it is the right driver to connect to the given URL but has trouble connecting to the database.

The java.util.Properties argument can be used to pass arbitrary string tag/value pairs as connection arguments. Normally at least "user" and "password" properties should be included in the Properties object.

**Parameters:**url - the URL of the database to which to connectinfo - 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 object that represents a connection to the URL **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### acceptsURL

boolean **acceptsURL**([String](http://docs.google.com/java/lang/String.html) url)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves whether the driver thinks that it can open a connection to the given URL. Typically drivers will return true if they understand the subprotocol specified in the URL and false if they do not.

**Parameters:**url - the URL of the database **Returns:**true if this driver understands the given URL; false otherwise **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### getPropertyInfo

[DriverPropertyInfo](http://docs.google.com/java/sql/DriverPropertyInfo.html)[] **getPropertyInfo**([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)

Gets information about the possible properties for this driver.

The getPropertyInfo method is intended to allow a generic GUI tool to discover what properties it should prompt a human for in order to get enough information to connect to a database. Note that depending on the values the human has supplied so far, additional values may become necessary, so it may be necessary to iterate though several calls to the getPropertyInfo method.

**Parameters:**url - the URL of the database to which to connectinfo - a proposed list of tag/value pairs that will be sent on connect open **Returns:**an array of DriverPropertyInfo objects describing possible properties. This array may be an empty array if no properties are required. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### getMajorVersion

int **getMajorVersion**()

Retrieves the driver's major version number. Initially this should be 1.

**Returns:**this driver's major version number

### getMinorVersion

int **getMinorVersion**()

Gets the driver's minor version number. Initially this should be 0.

**Returns:**this driver's minor version number

### jdbcCompliant

boolean **jdbcCompliant**()

Reports whether this driver is a genuine JDBC CompliantTM driver. A driver may only report true here if it passes the JDBC compliance tests; otherwise it is required to return false.

JDBC compliance requires full support for the JDBC API and full support for SQL 92 Entry Level. It is expected that JDBC compliant drivers will be available for all the major commercial databases.

This method is not intended to encourage the development of non-JDBC compliant drivers, but is a recognition of the fact that some vendors are interested in using the JDBC API and framework for lightweight databases that do not support full database functionality, or for special databases such as document information retrieval where a SQL implementation may not be feasible.

**Returns:**true if this driver is JDBC Compliant; false otherwise

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Driver.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/Date.html)   [**NEXT CLASS**](http://docs.google.com/java/sql/DriverManager.html) | [**FRAMES**](http://docs.google.com/index.html?java/sql/Driver.html)    [**NO FRAMES**](http://docs.google.com/Driver.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

[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).