**Module** [java.base](https://docs.oracle.com/en/java/javase/16/docs/api/java.base/module-summary.html)

**Package** [java.lang](https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/lang/package-summary.html)

**Since:** 1.7

# Interface AutoCloseable

----------------------------------------------------------------------------------------------------------------

**public interface AutoCloseable**

An object that may hold resources (such as file or socket handles) until it is closed. The [close()](https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/lang/AutoCloseable.html#close()) method of an AutoCloseable object is called automatically when exiting a try-with-resources block for which the object has been declared in the resource specification header. This construction ensures prompt release, avoiding resource exhaustion exceptions and errors that may otherwise occur.

**API Note:**

It is possible, and in fact common, for a base class to implement AutoCloseable even though not all of its subclasses or instances will hold releasable resources. For code that must operate in complete generality, or when it is known that the AutoCloseable instance requires resource release, it is recommended to use try-with-resources constructions. However, when using facilities such as [Stream](https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/util/stream/Stream.html) that support both I/O-based and non-I/O-based forms, try-with-resources blocks are in general unnecessary when using non-I/O-based forms.

## ***Methods of AutoCloseable Interface: -***

1. **void close () throws Exception: -** Closes this resource, relinquishing any underlying resources. This method is invoked automatically on objects managed by the try-with-resources statement.

**Module** [java.base](https://docs.oracle.com/en/java/javase/16/docs/api/java.base/module-summary.html)

**Package** [java.io](https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/io/package-summary.html)

**Since:** 1.5

# Interface Closeable

----------------------------------------------------------------------------------------------------------------

**public interface Closeable extends AutoCloseable**

A Closeable is a source or destination of data that can be closed. The close method is invoked to release resources that the object is holding (such as open files).

## ***Methods of Closeable Interface: -***

1. **void close () throws IOException: -** Closes this stream and releases any system resources associated with it. If the stream is already closed then invoking this method has no effect.

**Limitations of Closeable Interface**

Closeable throws only IOException and it cannot be changed without breaking legacy code. Therefore, AutoCloseable was introduced as it can throw an Exception.

**Closeable vs AutoCloseable**

* Closeable was introduced with JDK 5 whereas AutoCloseable was introduced with JDK 7+.
* Closeable extends AutoCloseable and Closeable is mainly directed to IO streams.
* Closeable extends IOException whereas AutoCloseable extends Exception.
* Closeable interface is idempotent (calling close () method more than once does not have any side effects) whereas AutoCloseable does not provide this feature.
* AutoCloseable was specially introduced to work with try-with-resources statements. Since Closeable implements AutoCloseable, therefore any class that implements Closeable also implements AutoCloseable interface and can use the try-with resources to close the files.