| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/javax/swing/undo/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/transaction/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/tools/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

## Package javax.tools

Provides interfaces for tools which can be invoked from a program, for example, compilers.

**See:**

[**Description**](#3znysh7)

| **Interface Summary** | |
| --- | --- |
| [**Diagnostic<S>**](http://docs.google.com/javax/tools/Diagnostic.html) | Interface for diagnostics from tools. |
| [**DiagnosticListener<S>**](http://docs.google.com/javax/tools/DiagnosticListener.html) | Interface for receiving diagnostics from tools. |
| [**FileObject**](http://docs.google.com/javax/tools/FileObject.html) | File abstraction for tools. |
| [**JavaCompiler**](http://docs.google.com/javax/tools/JavaCompiler.html) | Interface to invoke Java™ programming language compilers from programs. |
| [**JavaCompiler.CompilationTask**](http://docs.google.com/javax/tools/JavaCompiler.CompilationTask.html) | Interface representing a future for a compilation task. |
| [**JavaFileManager**](http://docs.google.com/javax/tools/JavaFileManager.html) | File manager for tools operating on Java™ programming language source and class files. |
| [**JavaFileManager.Location**](http://docs.google.com/javax/tools/JavaFileManager.Location.html) | Interface for locations of file objects. |
| [**JavaFileObject**](http://docs.google.com/javax/tools/JavaFileObject.html) | File abstraction for tools operating on Java™ programming language source and class files. |
| [**OptionChecker**](http://docs.google.com/javax/tools/OptionChecker.html) | Interface for recognizing options. |
| [**StandardJavaFileManager**](http://docs.google.com/javax/tools/StandardJavaFileManager.html) | File manager based on [java.io.File](http://docs.google.com/java/io/File.html). |
| [**Tool**](http://docs.google.com/javax/tools/Tool.html) | Common interface for tools that can be invoked from a program. |

| **Class Summary** | |
| --- | --- |
| [**DiagnosticCollector<S>**](http://docs.google.com/javax/tools/DiagnosticCollector.html) | Provides an easy way to collect diagnostics in a list. |
| [**ForwardingFileObject<F extends FileObject>**](http://docs.google.com/javax/tools/ForwardingFileObject.html) | Forwards calls to a given file object. |
| [**ForwardingJavaFileManager<M extends JavaFileManager>**](http://docs.google.com/javax/tools/ForwardingJavaFileManager.html) | Forwards calls to a given file manager. |
| [**ForwardingJavaFileObject<F extends JavaFileObject>**](http://docs.google.com/javax/tools/ForwardingJavaFileObject.html) | Forwards calls to a given file object. |
| [**SimpleJavaFileObject**](http://docs.google.com/javax/tools/SimpleJavaFileObject.html) | Provides simple implementations for most methods in JavaFileObject. |
| [**ToolProvider**](http://docs.google.com/javax/tools/ToolProvider.html) | Provides methods for locating tool providers, for example, providers of compilers. |

| **Enum Summary** | |
| --- | --- |
| [**Diagnostic.Kind**](http://docs.google.com/javax/tools/Diagnostic.Kind.html) | Kinds of diagnostics, for example, error or warning. |
| [**JavaFileObject.Kind**](http://docs.google.com/javax/tools/JavaFileObject.Kind.html) | Kinds of JavaFileObjects. |
| [**StandardLocation**](http://docs.google.com/javax/tools/StandardLocation.html) | Standard locations of file objects. |

## Package javax.tools Description

Provides interfaces for tools which can be invoked from a program, for example, compilers.

These interfaces and classes are required as part of the Java™ Platform, Standard Edition (Java SE), but there is no requirement to provide any tools implementing them.

Unless explicitly allowed, all methods in this package might throw a [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) if given a null argument or if given a [list or collection](http://docs.google.com/java/lang/Iterable.html) containing null elements. Similarly, no method may return null unless explicitly allowed.

This package is the home of the Java programming language compiler framework. This framework allows clients of the framework to locate and run compilers from programs. The framework also provides Service Provider Interfaces (SPI) for structured access to diagnostics ([DiagnosticListener](http://docs.google.com/javax/tools/DiagnosticListener.html)) as well as a file abstraction for overriding file access ([JavaFileManager](http://docs.google.com/javax/tools/JavaFileManager.html) and [JavaFileObject](http://docs.google.com/javax/tools/JavaFileObject.html)). See [JavaCompiler](http://docs.google.com/javax/tools/JavaCompiler.html) for more details on using the SPI.

There is no requirement for a compiler at runtime. However, if a default compiler is provided, it can be located using the [ToolProvider](http://docs.google.com/javax/tools/ToolProvider.html), for example:

JavaCompiler compiler = ToolProvider.getSystemJavaCompiler();

It is possible to provide alternative compilers or tools through the [service provider mechanism](http://docs.google.com/java/util/ServiceLoader.html).

For example, if com.vendor.VendorJavaCompiler is a provider of the JavaCompiler tool then its jar file would contain the file META-INF/services/javax.tools.JavaCompiler. This file would contain the single line:

com.vendor.VendorJavaCompiler

If the jar file is on the class path, VendorJavaCompiler can be located using code like this:

JavaCompiler compiler = ServiceLoader.load(JavaCompiler.class).iterator().next();

**Since:** 1.6

| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/javax/swing/undo/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/transaction/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/tools/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[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).