Overview Package Class Use Tree Deprecated Index Help
PREV NEXT FRAMES NO FRAMES All Classes

Java™ 2 Platform Standard Edition 5.0
API Specification

This document is the API specification for the Java 2 Platform Standard Edition 5.0.

See:
Description

Java 2 Platform Packages	
java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
java.awt.font	Provides classes and interface relating to fonts.
java.awt.geom	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
java.awt.im	Provides classes and interfaces for the input method framework.
java.awt.im.spi	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
java.awt.image	Provides classes for creating and modifying images.
java.awt.image.renderable	Provides classes and interfaces for producing rendering-independent images.
java.awt.print	Provides classes and interfaces for a general printing API.
java.beans	Contains classes related to developing *beans* -- components based on the JavaBeans™ architecture.
java.beans.beancontext	Provides classes and interfaces relating to bean context.
java.io	Provides for system input and output through data streams, serialization and the file system.
java.lang	Provides classes that are fundamental to the design of the Java programming language.
java.lang.annotation	Provides library support for the Java programming language annotation facility.
java.lang.instrument	Provides services that allow Java programming language agents to instrument programs running on the JVM.
java.lang.management	Provides the management interface for monitoring and management of the Java virtual machine as well as the operating system on which the Java virtual machine is running.
java.lang.ref	Provides reference-object classes, which support a limited degree of interaction with the garbage collector.
java.lang.reflect	Provides classes and interfaces for obtaining reflective information about classes and objects.
java.math	Provides classes for performing arbitrary-precision integer arithmetic (BigInteger) and arbitrary-precision decimal arithmetic (BigDecimal).
java.net	Provides the classes for implementing networking applications.
java.nio	Defines buffers, which are containers for data, and provides an overview of the other NIO packages.
java.nio.channels	Defines channels, which represent connections to entities that are capable of performing I/O operations, such as files and sockets; defines selectors, for multiplexed, non-blocking I/O operations.
java.nio.channels.spi	Service-provider classes for the `java.nio.channels` package.
java.nio.charset	Defines charsets, decoders, and encoders, for translating between bytes and Unicode characters.
java.nio.charset.spi	Service-provider classes for the `java.nio.charset` package.
java.rmi	Provides the RMI package.
java.rmi.activation	Provides support for RMI Object Activation.

java.rmi.dgc	Provides classes and interface for RMI distributed garbage-collection (DGC).
java.rmi.registry	Provides a class and two interfaces for the RMI registry.
java.rmi.server	Provides classes and interfaces for supporting the server side of RMI.
java.security	Provides the classes and interfaces for the security framework.
java.security.acl	The classes and interfaces in this package have been superseded by classes in the java.security package.
java.security.cert	Provides classes and interfaces for parsing and managing certificates, certificate revocation lists (CRLs), and certification paths.
java.security.interfaces	Provides interfaces for generating RSA (Rivest, Shamir and Adleman AsymmetricCipher algorithm) keys as defined in the RSA Laboratory Technical Note PKCS#1, and DSA (Digital Signature Algorithm) keys as defined in NIST's FIPS-186.
java.security.spec	Provides classes and interfaces for key specifications and algorithm parameter specifications.
java.sql	Provides the API for accessing and processing data stored in a data source (usually a relational database) using the Java^{TM} programming language.
java.text	Provides classes and interfaces for handling text, dates, numbers, and messages in a manner independent of natural languages.
java.util	Contains the collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes (a string tokenizer, a random-number generator, and a bit array).
java.util.concurrent	Utility classes commonly useful in concurrent programming.
java.util.concurrent.atomic	A small toolkit of classes that support lock-free thread-safe programming on single variables.
java.util.concurrent.locks	Interfaces and classes providing a framework for locking and waiting for conditions that is distinct from built-in synchronization and monitors.
java.util.jar	Provides classes for reading and writing the JAR (Java ARchive) file format, which is based on the standard ZIP file format with an optional manifest file.
java.util.logging	Provides the classes and interfaces of the Java^{TM} 2 platform's core logging facilities.
java.util.prefs	This package allows applications to store and retrieve user and system preference and configuration data.
java.util.regex	Classes for matching character sequences against patterns specified by regular expressions.
java.util.zip	Provides classes for reading and writing the standard ZIP and GZIP file formats.
javax.accessibility	Defines a contract between user-interface components and an assistive technology that provides access to those components.
javax.crypto	Provides the classes and interfaces for cryptographic operations.
javax.crypto.interfaces	Provides interfaces for Diffie-Hellman keys as defined in RSA Laboratories' PKCS #3.
javax.crypto.spec	Provides classes and interfaces for key specifications and algorithm parameter specifications.
javax.imageio	The main package of the Java Image I/O API.
javax.imageio.event	A package of the Java Image I/O API dealing with synchronous notification of events during the reading and writing of images.
javax.imageio.metadata	A package of the Java Image I/O API dealing with reading and writing metadata.
javax.imageio.plugins.bmp	Package containing the public classes used by the built-in BMP plug-in.
javax.imageio.plugins.jpeg	Classes supporting the built-in JPEG plug-in.
javax.imageio.spi	A package of the Java Image I/O API containing the plug-in interfaces for readers, writers, transcoders, and streams, and a runtime registry.
javax.imageio.stream	A package of the Java Image I/O API dealing with low-level I/O from files and streams.
javax.management	Provides the core classes for the Java Management Extensions.
javax.management.loading	Provides the classes which implement advanced dynamic loading.
javax.management.modelmbean	Provides the definition of the ModelMBean classes.
javax.management.monitor	Provides the definition of the monitor classes.
javax.management.openmbean	Provides the open data types and Open MBean descriptor classes.
javax.management.relation	Provides the definition of the Relation Service.
javax.management.remote	Interfaces for remote access to JMX MBean servers.
javax.management.remote.rmi	The RMI connector is a connector for the JMX Remote API that uses RMI to transmit client requests to a remote MBean server.
javax.management.timer	Provides the definition of the Timer MBean.
javax.naming	Provides the classes and interfaces for accessing naming services.

javax.naming.directory	Extends the `javax.naming` package to provide functionality for accessing directory services.
javax.naming.event	Provides support for event notification when accessing naming and directory services.
javax.naming.ldap	Provides support for LDAPv3 extended operations and controls.
javax.naming.spi	Provides the means for dynamically plugging in support for accessing naming and directory services through the `javax.naming` and related packages.
javax.net	Provides classes for networking applications.
javax.net.ssl	Provides classes for the secure socket package.
javax.print	Provides the principal classes and interfaces for the Java™ Print Service API.
javax.print.attribute	Provides classes and interfaces that describe the types of Java™ Print Service attributes and how they can be collected into attribute sets.
javax.print.attribute.standard	Package javax.print.attribute.standard contains classes for specific printing attributes.
javax.print.event	Package javax.print.event contains event classes and listener interfaces.
javax.rmi	Contains user APIs for RMI-IIOP.
javax.rmi.CORBA	Contains portability APIs for RMI-IIOP.
javax.rmi.ssl	Provides implementations of `RMIClientSocketFactory` and `RMIServerSocketFactory` over the Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocols.
javax.security.auth	This package provides a framework for authentication and authorization.
javax.security.auth.callback	This package provides the classes necessary for services to interact with applications in order to retrieve information (authentication data including usernames or passwords, for example) or to display information (error and warning messages, for example).
javax.security.auth.kerberos	This package contains utility classes related to the Kerberos network authentication protocol.
javax.security.auth.login	This package provides a pluggable authentication framework.
javax.security.auth.spi	This package provides the interface to be used for implementing pluggable authentication modules.
javax.security.auth.x500	This package contains the classes that should be used to store X500 Principal and X500 Private Crendentials in a *Subject*.
javax.security.cert	Provides classes for public key certificates.
javax.security.sasl	Contains class and interfaces for supporting SASL.
javax.sound.midi	Provides interfaces and classes for I/O, sequencing, and synthesis of MIDI (Musical Instrument Digital Interface) data.
javax.sound.midi.spi	Supplies interfaces for service providers to implement when offering new MIDI devices, MIDI file readers and writers, or sound bank readers.
javax.sound.sampled	Provides interfaces and classes for capture, processing, and playback of sampled audio data.
javax.sound.sampled.spi	Supplies abstract classes for service providers to subclass when offering new audio devices, sound file readers and writers, or audio format converters.
javax.sql	Provides the API for server side data source access and processing from the Java™ programming language.
javax.sql.rowset	Standard interfaces and base classes for JDBC `RowSet` implementations.
javax.sql.rowset.serial	Provides utility classes to allow serializable mappings between SQL types and data types in the Java programming language.
javax.sql.rowset.spi	The standard classes and interfaces that a third party vendor has to use in its implementation of a synchronization provider.
javax.swing	Provides a set of "lightweight" (all-Java language) components that, to the maximum degree possible, work the same on all platforms.
javax.swing.border	Provides classes and interface for drawing specialized borders around a Swing component.
javax.swing.colorchooser	Contains classes and interfaces used by the `JColorChooser` component.
javax.swing.event	Provides for events fired by Swing components.
javax.swing.filechooser	Contains classes and interfaces used by the `JFileChooser` component.
javax.swing.plaf	Provides one interface and many abstract classes that Swing uses to provide its pluggable look-and-feel capabilities.
javax.swing.plaf.basic	Provides user interface objects built according to the Basic look and feel.
javax.swing.plaf.metal	Provides user interface objects built according to the Java look and feel (once codenamed *Metal*), which is the default look and feel.
javax.swing.plaf.multi	Provides user interface objects that combine two or more look and feels.
javax.swing.plaf.synth	Synth is a skinnable look and feel in which all painting is delegated.
javax.swing.table	Provides classes and interfaces for dealing with `javax.swing.JTable`.

javax.swing.text	Provides classes and interfaces that deal with editable and noneditable text components.
javax.swing.text.html	Provides the class `HTMLEditorKit` and supporting classes for creating HTML text editors.
javax.swing.text.html.parser	Provides the default HTML parser, along with support classes.
javax.swing.text.rtf	Provides a class (`RTFEditorKit`) for creating Rich-Text-Format text editors.
javax.swing.tree	Provides classes and interfaces for dealing with `javax.swing.JTree`.
javax.swing.undo	Allows developers to provide support for undo/redo in applications such as text editors.
javax.transaction	Contains three exceptions thrown by the ORB machinery during unmarshalling.
javax.transaction.xa	Provides the API that defines the contract between the transaction manager and the resource manager, which allows the transaction manager to enlist and delist resource objects (supplied by the resource manager driver) in JTA transactions.
javax.xml	Defines core XML constants and functionality from the XML specifications.
javax.xml.datatype	XML/Java Type Mappings.
javax.xml.namespace	XML Namespace processing.
javax.xml.parsers	Provides classes allowing the processing of XML documents.
javax.xml.transform	This package defines the generic APIs for processing transformation instructions, and performing a transformation from source to result.
javax.xml.transform.dom	This package implements DOM-specific transformation APIs.
javax.xml.transform.sax	This package implements SAX2-specific transformation APIs.
javax.xml.transform.stream	This package implements stream- and URI- specific transformation APIs.
javax.xml.validation	This package provides an API for validation of XML documents.
javax.xml.xpath	This package provides an *object-model neutral* API for the evaluation of XPath expressions and access to the evaluation environment.
org.ietf.jgss	This package presents a framework that allows application developers to make use of security services like authentication, data integrity and data confidentiality from a variety of underlying security mechanisms like Kerberos, using a unified API.
org.omg.CORBA	Provides the mapping of the OMG CORBA APIs to the Java^{TM} programming language, including the class `ORB`, which is implemented so that a programmer can use it as a fully-functional Object Request Broker (ORB).
org.omg.CORBA_2_3	The CORBA_2_3 package defines additions to existing CORBA interfaces in the Java 2 Standard Edition. These changes occurred in recent revisions to the CORBA API defined by the OMG. The new methods were added to interfaces derived from the corresponding interfaces in the CORBA package. This provides backward compatibility and avoids breaking the JCK tests.
org.omg.CORBA_2_3.portable	Provides methods for the input and output of value types, and contains other updates to the `org/omg/CORBA/portable` package.
org.omg.CORBA.DynAnyPackage	Provides the exceptions used with the `DynAny` interface (`InvalidValue`, `Invalid`, `InvalidSeq`, and `TypeMismatch`).
org.omg.CORBA.ORBPackage	Provides the exception `InvalidName`, which is thrown by the method `ORB.resolve_initial_references` and the exception `InconsistentTypeCode`, which is thrown by the Dynamic Any creation methods in the ORB class.
org.omg.CORBA.portable	Provides a portability layer, that is, a set of ORB APIs that makes it possible for code generated by one vendor to run on another vendor's ORB.
org.omg.CORBA.TypeCodePackage	Provides the user-defined exceptions `BadKind` and `Bounds`, which are thrown by methods in in the class `TypeCode`.
org.omg.CosNaming	Provides a naming service for Java IDL.
org.omg.CosNaming.NamingContextExtPackage	This package contains the following classes, which are used in `org.omg.CosNaming.NamingContextExt`:
org.omg.CosNaming.NamingContextPackage	This package contains Exception classes for the `org.omg.CosNaming` package.
org.omg.Dynamic	This package contains the `Dynamic` module specified in the OMG Portable Interceptor specification, http://cgi.omg.org/cgi-bin/doc?ptc/2000-08-06, section 21.9.
org.omg.DynamicAny	Provides classes and interfaces that enable traversal of the data value associated with an `any` at runtime, and extraction of the primitive constituents of the data value.
org.omg.DynamicAny.DynAnyFactoryPackage	This package contains classes and exceptions from the `DynAnyFactory` interface of the `DynamicAny` module specified in the OMG *The Common Object Request Broker: Architecture and Specification*, http://cgi.omg.org/cgi-bin/doc?formal/99-10-07, section 9.2.2.
org.omg.DynamicAny.DynAnyPackage	This package contains classes and exceptions from the `DynAny` interface of the `DynamicAny` module specified in the OMG *The Common Object Request Broker: Architecture and Specification*, http://cgi.omg.org/cgi-bin/doc?formal/99-10-07, section 9.2.

org.omg.IOP	This package contains the `IOP` module specified in the OMG document *The Common Object Request Broker: Architecture and Specification*, http://cgi.omg.org/cgi-bin/doc?formal/99-10-07, section 13.6.
org.omg.IOP.CodecFactoryPackage	This package contains the exceptions specified in the IOP::CodeFactory interface (as part of the Portable Interceptors spec).
org.omg.IOP.CodecPackage	This package is generated from the IOP::Codec IDL interface definition.
org.omg.Messaging	This package contains the `Messaging` module specified in the OMG CORBA Messaging specification, http://cgi.omg.org/cgi-bin/doc?formal/99-10-07.
org.omg.PortableInterceptor	Provides a mechanism to register ORB hooks through which ORB services can intercept the normal flow of execution of the ORB.
org.omg.PortableInterceptor.ORBInitInfoPackage	This package contains the exceptions and typedefs from the `ORBInitInfo` local interface of the `PortableInterceptor` module specified in the OMG Portable Interceptor specification, http://cgi.omg.org/cgi-bin/doc?ptc/2000-08-06, section 21.7.2.
org.omg.PortableServer	Provides classes and interfaces for making the server side of your applications portable across multivendor ORBs.
org.omg.PortableServer.CurrentPackage	Provides method implementations with access to the identity of the object on which the method was invoked.
org.omg.PortableServer.POAManagerPackage	Encapsulates the processing state of the POAs it is associated with.
org.omg.PortableServer.POAPackage	Allows programmers to construct object implementations that are portable between different ORB products.
org.omg.PortableServer.portable	Provides classes and interfaces for making the server side of your applications portable across multivendor ORBs.
org.omg.PortableServer.ServantLocatorPackage	Provides classes and interfaces for locating the servant.
org.omg.SendingContext	Provides support for the marshalling of value types.
org.omg.stub.java.rmi	Contains RMI-IIOP Stubs for the Remote types that occur in the `java.rmi` package.
org.w3c.dom	Provides the interfaces for the Document Object Model (DOM) which is a component API of the Java API for XML Processing.
org.w3c.dom.bootstrap	
org.w3c.dom.events	
org.w3c.dom.ls	
org.xml.sax	This package provides the core SAX APIs.
org.xml.sax.ext	This package contains interfaces to SAX2 facilities that conformant SAX drivers won't necessarily support.
org.xml.sax.helpers	This package contains "helper" classes, including support for bootstrapping SAX-based applications.

Other Packages	
javax.activity	Contains Activity service related exceptions thrown by the ORB machinery during unmarshalling.

This document is the API specification for the Java 2 Platform Standard Edition 5.0.

Overview Package Class Use **Tree** **Deprecated** **Index** **Help**
PREV NEXT **FRAMES** **NO FRAMES** **All Classes**

Java™ 2 Platform
Standard Ed. 5.0

Submit a bug or feature
For further API reference and developer documentation, see Java 2 SDK SE Developer Documentation. That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.