How This API Document Is Organized

This API (Application Programming Interface) document has pages corresponding to the items in the navigation bar, described as follows.

Package

Each package has a page that contains a list of its classes and interfaces, with a summary for each. These pages may contain six categories:

- Interfaces
- Classes
- Enums
- Exceptions
- Errors
- Annotation Types

Class or Interface

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three sections consisting of a class/interface description, summary tables, and detailed member descriptions:

- Class Inheritance Diagram
- Direct Subclasses
- All Known Subinterfaces
- All Known Implementing Classes
- Class or Interface Declaration
- Class or Interface Description

- Nested Class Summary
- Field Summary
- Property Summary
- Constructor Summary
- Method Summary

- Field Details
- Property Details
- Constructor Details
- Method Details

The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

Annotation Type

Each annotation type has its own separate page with the following sections:

- Annotation Type Declaration
- Annotation Type Description
- Required Element Summary
- Optional Element Summary
- Element Details

Enum

Each enum has its own separate page with the following sections:

- Enum Declaration
- Enum Description
- Enum Constant Summary

- Enum Constant Details

Use

Each documented package, class and interface has its own Use page. This page describes what packages, classes, methods, constructors and fields use any part of the given class or package. Given a class or interface A, its "Use" page includes subclasses of A, fields declared as A, methods that return A, and methods and constructors with parameters of type A. You can access this page by first going to the package, class or interface, then clicking on the "Use" link in the navigation bar.

Tree (Class Hierarchy)

There is a Class Hierarchy page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. Classes are organized by inheritance structure starting with `java.lang.Object`. Interfaces do not inherit from `java.lang.Object`.

- When viewing the Overview page, clicking on "Tree" displays the hierarchy for all packages.
- When viewing a particular package, class or interface page, clicking on "Tree" displays the hierarchy for only that package.

Deprecated API

The Deprecated API page lists all of the API that have been deprecated. A deprecated API is not recommended for use, generally due to shortcomings, and a replacement API is usually given. Deprecated APIs may be removed in future implementations.

Index

The Index contains an alphabetic index of all classes, interfaces, constructors, methods, and fields, as well as lists of all packages and all classes.

Serialized Form

Each serializable or externalizable class has a description of its serialization fields and methods. This information is of interest to those who implement rather than use the API. While there is no link in the navigation bar, you can get to this information by going to any serialized class and clicking "Serialized Form" in the "See Also" section of the class description.

Constant Field Values

The Constant Field Values page lists the static final fields and their values.

Search

You can search for definitions of modules, packages, types, fields, methods, system properties and other terms defined in the API, using some or all of the name, optionally using "camel-case" abbreviations. For example:

- `j.l.obj` will match "java.lang.Object"
- `InpStr` will match "java.io.InputStream"
- `HM.cK` will match "java.util.HashMap.containsKey(Object)"

Refer to the Javadoc Search Specification for a full description of search features.

This help file applies to API documentation generated by the standard doclet.