| | [**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/rmi/ssl/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/security/auth/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/script/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

## Package javax.script

The scripting API consists of interfaces and classes that define Java TM Scripting Engines and provides a framework for their use in Java applications.

**See:**

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

| **Interface Summary** | |
| --- | --- |
| [**Bindings**](http://docs.google.com/javax/script/Bindings.html) | A mapping of key/value pairs, all of whose keys are Strings. |
| [**Compilable**](http://docs.google.com/javax/script/Compilable.html) | The optional interface implemented by ScriptEngines whose methods compile scripts to a form that can be executed repeatedly without recompilation. |
| [**Invocable**](http://docs.google.com/javax/script/Invocable.html) | The optional interface implemented by ScriptEngines whose methods allow the invocation of procedures in scripts that have previously been executed. |
| [**ScriptContext**](http://docs.google.com/javax/script/ScriptContext.html) | The interface whose implementing classes are used to connect Script Engines with objects, such as scoped Bindings, in hosting applications. |
| [**ScriptEngine**](http://docs.google.com/javax/script/ScriptEngine.html) | ScriptEngine is the fundamental interface whose methods must be fully functional in every implementation of this specification. |
| [**ScriptEngineFactory**](http://docs.google.com/javax/script/ScriptEngineFactory.html) | ScriptEngineFactory is used to describe and instantiate ScriptEngines. |

| **Class Summary** | |
| --- | --- |
| [**AbstractScriptEngine**](http://docs.google.com/javax/script/AbstractScriptEngine.html) | Provides a standard implementation for several of the variants of the eval method. |
| [**CompiledScript**](http://docs.google.com/javax/script/CompiledScript.html) | Extended by classes that store results of compilations. |
| [**ScriptEngineManager**](http://docs.google.com/javax/script/ScriptEngineManager.html) | The ScriptEngineManager implements a discovery and instantiation mechanism for ScriptEngine classes and also maintains a collection of key/value pairs storing state shared by all engines created by the Manager. |
| [**SimpleBindings**](http://docs.google.com/javax/script/SimpleBindings.html) | A simple implementation of Bindings backed by a HashMap or some other specified Map. |
| [**SimpleScriptContext**](http://docs.google.com/javax/script/SimpleScriptContext.html) | Simple implementation of ScriptContext. |

| **Exception Summary** | |
| --- | --- |
| [**ScriptException**](http://docs.google.com/javax/script/ScriptException.html) | The generic Exception class for the Scripting APIs. |

## Package javax.script Description

The scripting API consists of interfaces and classes that define Java TM Scripting Engines and provides a framework for their use in Java applications. This API is intended for use by application programmers who wish to execute programs written in scripting languages in their Java applications. The scripting language programs are usually provided by the end-users of the applications.

The main areas of functionality of javax.script package include

1. **Script execution**: Scripts are streams of characters used as sources for programs executed by script engines. Script execution uses [eval](http://docs.google.com/javax/script/ScriptEngine.html#eval(java.lang.String,%20javax.script.ScriptContext)) methods of [ScriptEngine](http://docs.google.com/javax/script/ScriptEngine.html) and methods of the [Invocable](http://docs.google.com/javax/script/Invocable.html) interface.
2. **Binding**: This facility allows Java objects to be exposed to script programs as named variables. [Bindings](http://docs.google.com/javax/script/Bindings.html) and [ScriptContext](http://docs.google.com/javax/script/ScriptContext.html) classes are used for this purpose.
3. **Compilation**: This functionality allows the intermediate code generated by the front-end of a script engine to be stored and executed repeatedly. This benefits applications that execute the same script multiple times. These applications can gain efficiency since the engines' front-ends only need to execute once per script rather than once per script execution. Note that this functionality is optional and script engines may choose not to implement it. Callers need to check for availability of the [Compilable](http://docs.google.com/javax/script/Compilable.html) interface using an *instanceof* check.
4. **Invocation**: This functionality allows the reuse of intermediate code generated by a script engine's front-end. Whereas Compilation allows entire scripts represented by intermediate code to be re-executed, Invocation functionality allows individual procedures/methods in the scripts to be re-executed. As in the case with compilation, not all script engines are required to provide this facility. Caller has to check for [Invocable](http://docs.google.com/javax/script/Invocable.html) availability.
5. **Script engine discovery and Metadata**: Applications written to the Scripting API might have specific requirements on script engines. Some may require a specific scripting language and/or version while others may require a specific implementation engine and/or version. Script engines are packaged in a specified way so that engines can be discovered at runtime and queried for attributes. The Engine discovery mechanism is based on the Service discovery mechanism described in the **Jar File Specification**. Script engine implementing classes are packaged in jar files that include a text resource named **META-INF/services/javax.script.ScriptEngineFactory**. This resource must include a line for each [ScriptEngineFactory](http://docs.google.com/javax/script/ScriptEngineFactory.html) that is packaged in the jar file. [ScriptEngineManager](http://docs.google.com/javax/script/ScriptEngineManager.html) includes [getEngineFactories](http://docs.google.com/javax/script/ScriptEngineManager.html#getEngineFactories()) method to get all [ScriptEngineFactory](http://docs.google.com/javax/script/ScriptEngineFactory.html) instances discovered using this mechanism. ScriptEngineFactory has methods to query attributes about script engine.

**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/rmi/ssl/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/security/auth/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/script/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).